

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Documentation de bases de données créées lors d'une enquête par questionnaire

Aupaix, Marie-Christine; Roelandt, Eric

Award date:
1991

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Notre Dame de la Paix de Namur

DOCUMENTATION DE BASES DE DONNEES
CREEES LORS D'UNE ENQUETE PAR
QUESTIONNAIRE

AUPAIX Marie-Christine
ROELANDT Eric

Promoteur Madame M. NOIRHOMME

Institut d'Informatique

21, rue Grandgagnage
5000 NAMUR (BELGIQUE)

Année Académique 1990-1991

Mémoire présenté pour l'obtention du diplôme
de Licencié et Maître en Sciences Informatiques

AVANT PROPOS

Que les premières lignes de ce mémoire soient notre meilleur ambassadeur pour adresser nos plus vifs remerciements aux personnes qui nous ont aidés à l'élaboration et à la rédaction de cette étude et que nous désirons citer maintenant.

Madame Monique NOIRHOMME, Promoteur de notre mémoire, qui a accepté d'en assurer la direction et qui nous a conseillés tout au long de cette année académique,

Madame Joanne LAMB, Chef du département informatique du Centre for Educational Sociology de l'Université d'Edimbourg, pour son aimable collaboration et ses nombreux conseils,

L'ensemble du personnel du Centre for Educational Sociology pour l'accueil qu'il nous a réservés lors de notre stage en Ecosse ainsi que pour leur précieuse documentation.

Nos remerciements s'adressent enfin à Mademoiselle Anne Vandenbroucke qui nous a aidés par une lecture critique de notre mémoire, ses conseils et sa vigilance nous ont été d'une aide précieuse.

A chacun, nous exprimons notre profonde reconnaissance.

ABSTRACT

The Centre for Educational Sociology (CES) of Edinburgh University is concerned with making surveys about education training. All these surveys, carried out for 10 years in CES, have led to a huge storage of data relative to this specific research domain. The thesis object consist in determining a database documentation system, which will help people who want to consult the data and who don't know about the context of the database design, to access easily these databases.

RESUME

Le Centre for Educational Sociology (CES) de l'Université d'Edimbourg réalise diverses enquêtes dont le thème central est l'éducation. L'ensemble de ces enquêtes, effectué depuis 10 ans, a permis de rassembler une multitude de données relatives à ce domaine de recherche. L'objet du mémoire consiste avant tout à établir un système de documentation des bases de données ainsi générées, de manière à en organiser un accès facile afin que toute personne désirant accéder aux informations stockées dans ces bases et ne connaissant pas le contexte lié à leur conception, y parvienne aisément.

0. TABLE DES MATIERES

0. Table des matières	1
1. Introduction	4
2. Contexte de travail	
2.1 La description du Centre for Educational Sociology	6
2.1.1 Qu'est-ce que le CES ?	6
2.1.2 Le programme de recherche du CES	7
2.1.3 La méthode de travail du CES	9
2.2 Les logiciels générateurs de questionnaires ...	15
2.2.1 Le traitement d'enquêtes par le système BLAISE ..	15
2.2.2 Le système SNAP	16
2.2.3 Le système QUESTMAST	17
2.3 L'élaboration d'un système de documentation ...	18
2.3.1 Qu'est-ce qu'une base de données documentation ?	18
2.3.2 Les objectifs d'une base de données documentation	18
2.3.3 Le prototype d'une base de données documentation : DOCDB	19
3. Considérations théoriques	
3.1 Les phases générales d'une enquête par questionnaire	23
3.1.1 Détermination des objectifs et évaluation des moyens d'une enquête	23
3.1.2 Définition de l'échantillon	24
3.1.3 Rédaction du questionnaire	25
3.1.4 Dépouillement et codage des données	30
3.1.5 Analyse et interprétation des résultats ..	31

3.2 Les méthodes d'échantillonnage	32
3.2.1 Les notions	32
3.2.2 La taille de l'échantillon	33
3.2.3 Les méthodes d'échantillonnage	36
4. Les éléments à prendre en considération pour l'élaboration des modèles	
4.1 La définition d'une question	42
4.2 Le progiciel QUESTMAST	44
5. Description des modèles d'analyse fonctionnelle du système d'information	
5.1 Le modèle général des opérations de traitement des informations d'une enquête par questionnaire	53
5.2 Les modèles de traitement des informations	56
5.2.1 La dynamique des traitements	56
5.2.2 Le diagramme des flux	57
5.3 Le modèle Entité/Association des informations	59
5.3.1 Les entités	59
5.3.2 Les relations	68
5.3.3 Les schémas et sous-schémas	70
5.4 Le modèle MAG du système d'oinformations	75
5.4.1 Les relations	76
5.3.2 Les schémas	79
6. La documentation des variables	
6.1 L'intégration du programme DOCPROJ dans le projet METADATA database	95

6.2 La spécification du programme DOCPROJ	96
6.2.1 Détermination du schéma Entité/ Association	96
6.2.2 Implémentation de la documentation des variables	97
7. Conclusion	103

Annexes

- Annexe 1 : Exemple de questionnaire
- Annexe 2 : Significations du schéma Entité/Association
- Annexe 3 : Schéma SIR de la base de données DOCPROJ
- Annexe 4 : Spécifications des variables du programme DOCPROJ
- Annexe 5 : Structure et listing des programmes PQL de DOCPROJ
- Annexe 6 : Application de la notion de raisonnement déductif

Bibliographie

1. INTRODUCTION

L'objectif du stage de troisième licence est de confronter les étudiants à un environnement différent de celui de la faculté, un environnement plus proche de la réalité du milieu professionnel.

En ce qui nous concerne, nous avons effectué ce stage à l'Université d'Edimbourg, plus précisément au "Centre for Educational Sociology", CES.

Ce mémoire a pour ambition de vous présenter ce centre, ses activités, son environnement de recherche et le travail que nous avons été amenés à réaliser au sein de cet organisme.

Ainsi, dans un premier temps, afin de mieux situer le travail que nous avons réalisé au CES, nous nous proposons dans le cadre du chapitre 2, de décrire différents aspects des activités exercées par ce centre dont, entre autres, la méthodologie et l'objet du travail de recherche, les outils de cette recherche, à savoir les enquêtes, l'environnement informatique du CES, les bases de données générées par le centre ainsi que le projet de documentation de celles-ci.

Certains aspects plus théoriques relatifs à la méthodologie des enquêtes par questionnaire seront explicités dans le chapitre 3. Cet exposé s'avère nécessaire afin que soit clairement compris et appréhendé le cas particulier retenu dans le cadre de ce mémoire : le CES.

De plus, afin d'en cerner toutes les spécificités, d'autres éléments clés devront encore être précisés dans le chapitre 4. Il s'agit d'une part, du concept de question, tel qu'il est vu par le CES et d'autre part, du progiciel QUESTMAST, créé et développé par le CES afin d'aider à la conception de questionnaires.

Après avoir pris connaissance du chapitre 2, 3 et 4, s'étant familiarisé avec le monde de l'enquête, et plus particulièrement de l'enquête par questionnaire, avec le CES et son mode de travail, le lecteur armé de ces différents concepts

peut appréhender, dans les chapitres qui suivent, la description du travail que nous avons réalisé au CES.

Deux projets nous ont été impartis. Les responsables du département informatique nous ont impliqué dans la création d'un programme dont l'objectif était de combler certaines lacunes de la phase d'encodage. Comme on peut s'en rendre compte, ce point ne représente qu'une part infime du travail réalisé dans le centre ; il était la réponse à une demande ponctuelle de programmation émanant des membres du CES.

Le "gros" du projet consistait en la conception d'un programme de documentation. La documentation d'une base de données est nécessaire pour obtenir une utilisation facile de celle-ci par une personne étrangère au projet de recherche, qui ne connaît donc pas nécessairement le contexte lié à la conception de cette base.

Dans un premier temps, nous avons réalisé l'analyse fonctionnelle du système d'information que représente le processus d'une enquête par questionnaire (Schéma Entité/Association, modèle de la dynamique des traitements, modèle d'Accès Généralisé). Dans un second temps, nous nous sommes attachés à élaborer le programme de documentation relatif à un aspect particulier de l'enquête, les variables associées aux questions.

2. CONTEXTE DE TRAVAIL

2.1 La description du Centre for Educational Sociology

(CES)

2.1.1 Qu'est-ce que le CES ?

Le CES a été reconnu en 1972 par l'Université d'Edinburgh et a été financé de manière continue par le Scottish Education Department (SED) depuis sa fondation et par le Economic and Social Research Council (ESRC) depuis 1974. Le centre est devenu un Designated Research Centre (DRC) du ESRC en janvier 1987.

Les autres sources de financement de cette institution sont l'agence pour la formation (Training Agency), le département de l'industrie pour l'Ecosse, le département pour l'emploi, les autorités locales et certaines fondations privées du Royaume-Uni ou de l'étranger¹.

Les enquêtes et études réalisées au sein de ce centre sont menées par des chercheurs rémunérés à temps plein ou non, appartenant au CES ou par des associés ou d'autres chercheurs travaillant pour d'autres universités à l'étranger, comme au Royaume-Uni.

En automne 1990, il y avait environ une vingtaine d'employés travaillant à temps plein ou à temps partiel au CES, y compris les deux directeurs et d'autres personnes travaillant pour l'Université d'Edinburgh ou pour d'autres universités.

La décomposition fonctionnelle du personnel du CES est la suivante :

- 5 personnes dont une à mi-temps s'occupant de la mise en oeuvre de l'enquête sur le terrain, assistées de temporaires pour l'encodage des données ;

¹ Inspiré de "The DRC in the CES" : Rapport au ESRC des activités du CES (1987-1989).

- 6 analystes-programmeurs ;
- 5 personnes se consacrant aux charges de secrétariat et d'administration, dont deux sont à temps partiel ;
- un administrateur de recherche ;
- 6 membres du personnel académique ou chercheurs, incluant les directeurs.

Cette classification n'est pas stricte, mais elle reflète la diversité du personnel nécessaire pour mettre sur pied et mener à bien l'ensemble du programme de recherche qui est imparti au CES.

2.1.2 Le programme de recherche du CES

La recherche est un secteur du CES extrêmement important. En effet, la plupart des financements dont bénéficie le centre, provient d'institutions et de sociétés désireuses de réaliser une étude précise et bien définie dont le thème central est l'éducation.

Dès lors, l'objet principal de la recherche menée par le CES consiste en l'étude des expériences des jeunes en matière de formation scolaire, mais encore leur positionnement sur le marché de l'emploi, leur situation familiale et aussi, l'influence quant à ces expériences, d'une part des différents programmes et interventions politiques et d'autre part, des changements sociaux inattendus.

Voici les domaines de recherche du CES:

- l'environnement social dans l'éducation et le marché du travail ;
- l'organisation et l'efficacité du système éducatif ;
- le marché du travail pour les jeunes et la formation après l'enseignement obligatoire ;
- l'évaluation des initiatives politiques ;
- l'évaluation spatiale des résultats ;

- les méthodes et concepts à multi-niveaux ;
- les méthodes qualitatives.

Les études à caractère sociologique sont généralement soumises aux changements ; ceux-ci ont un impact important sur les éléments qui font l'objet de ces études et cela à plusieurs points de vue.

Ainsi en Ecosse, pour le cas qui nous concerne, afin d'appréhender efficacement la situation des jeunes face aux choix scolaires et le degré de corrélation qui existe entre ceux-ci et leur insertion sociale et professionnelle, il s'avère nécessaire de tenir compte de ces éléments à caractère changeant qui prennent de plus en plus d'ampleur. Sur le plan politique, le pays a connu une nouvelle législation en matière d'enseignement qui a modifié l'ensemble de l'organisation du système éducatif. Sur le plan social, la polarisation sociale et d'autres distorsions n'ont fait que s'accroître au fil des années. D'un autre point de vue, on notera encore que diverses améliorations méthodologiques dans la conception et l'analyse en recherche sociologique, ont été opérées, non seulement au sein même du CES, mais encore grâce à la contribution de personnes externes.

Le centre désire développer son action autour de quatre axes ; c'est pourquoi divers progrès significatifs ont déjà pu être observés.

En premier lieu, les membres du CES pensent qu'ils peuvent apporter une contribution non négligeable dans la conception et dans l'évaluation des différentes politiques menées par les autorités, responsables de l'éducation en Ecosse.

Ceci implique, en deuxième lieu, d'évaluer les changements sociaux résultant de ces multiples interventions politiques.

En troisième lieu, dans ce type d'études, il s'avère nécessaire de raisonner dans un cadre théorique relativement large ; en effet, ce projet de recherche requiert une connaissance approfondie d'une part, des théories sociales et d'autre part, des méthodologies à appliquer.

Et, finalement, la quatrième préoccupation se rapporte à la frontière fluctuante entre l'administration du bien public et la pratique de la science sociale.

Pour réaliser ces quatre objectifs, le CES a l'intention d'étendre ses séries d'enquêtes et également d'améliorer ses méthodes de gestion et d'analyse des données.

2.1.3 La méthode de travail du CES

Les enquêtes types du CES

Pour effectuer ses recherches, le CES recourt à diverses enquêtes grâce auxquelles est emmagasinée une multitude de données. Avant de commencer la description de ces enquêtes, précisons quelques éléments spécifiques au contexte de travail du CES.

L'enquête principale de ce centre est appelée "The Scottish Young People's Survey" ou encore SYPS. Il s'agit d'une enquête postale qui est envoyée biannuellement (au printemps et en automne) à un échantillon représentant environ 10% des jeunes écossais. Le schéma ci-dessous décrit l'échantillon visé par cette enquête. Nous ferons référence à ce graphe à plusieurs reprises au cours de cette section.

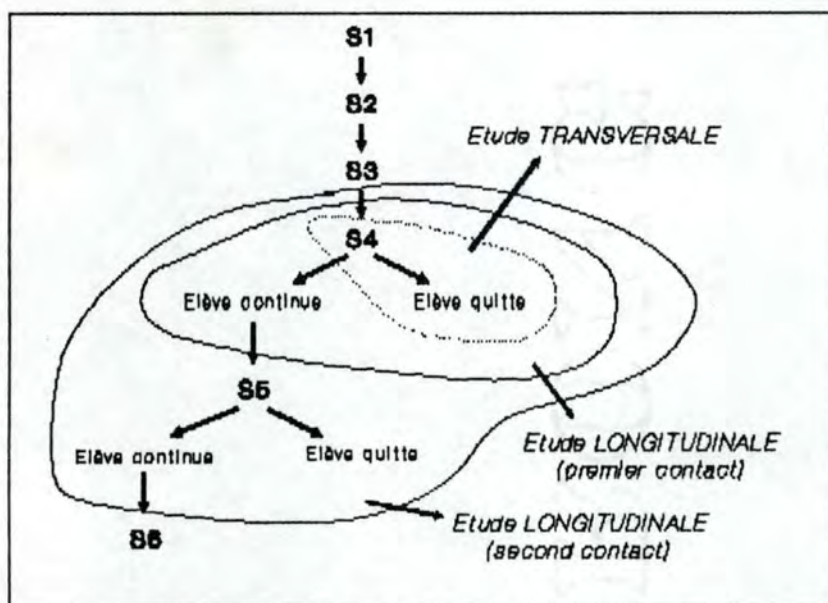


Fig. 2.11 : Représentation des échantillons des enquêtes

Cette enquête combine deux études distinctes.

La première définit une enquête TRANSVERSALE ("cross-sectional") dont l'échantillon porte sur les élèves qui quittent l'école après leur S4². Cette étude, menée depuis 1977, est envoyée une année sur deux, au cours du mois de mars. En effet, les chercheurs du CES ont décidé de prendre contact avec ces jeunes, neuf mois après qu'ils aient quitté l'école, ceci afin de laisser le temps aux jeunes de s'établir professionnellement ou de s'inscrire dans un programme de formation.

La seconde étude, menée depuis 1985, définit une enquête LONGITUDINALE ("cohort")³ dont l'échantillon est composé des élèves qui terminent leur S4, indépendamment du fait qu'ils continuent ou quittent l'école⁴. Cette étude est également envoyée une année sur deux. Le premier contact s'effectue aussi neuf mois après la fin de leur S4. Mais un second contact est établi deux ans et demi après la fin de leur S4, soit neuf mois après l'équivalent de la fin du cycle secondaire (S6). Les individus, interrogés lors de ce second contact, appartiennent à une des catégories suivantes :

1. ceux qui ont cessé leurs études après S4
2. ceux qui ont cessé leurs études après S5
3. ceux qui ont cessé leurs études après S6
4. ceux qui ont terminé leur S6, et qui commencent d'autres études

Pour les trois premières catégories, il faut encore distinguer ceux qui entrent directement dans la vie professionnelle et ceux qui décident de suivre des programmes de formation.

Le but de ces études est d'étudier et de comparer la situation professionnelle des jeunes qui ont quitté l'école "prématurément" et ceux qui ont achevé le cycle secondaire. D'après les résultats obtenus, le CES pourra mettre en

2 La fin de la S4 correspond à la fin de notre cycle secondaire inférieur.

3 Par la suite, nous nous intéresserons uniquement à ce type d'études.

4 On observe un chevauchement entre les individus qui composent les deux échantillons (cfr Fig.2.1.1 : ... et --).

évidence les problèmes du marché de travail et souligner les améliorations à apporter dans les programmes de formation.

Toute la recherche du CES ne consiste pas uniquement dans les SYPS, mais cette série d'enquêtes donne une cohésion au travail du CES et confère une base empirique pour une intégration théorique de différents domaines et disciplines.

Comme il a été dit précédemment, le thème principal est l'éducation dans sa généralité. Il n'est dès lors pas surprenant de constater que les autres études menées au CES concernent toujours le système éducatif et adoptent les mécanismes développés pour les SYPS. A titre d'information, voici un exemple d'étude commanditée par un organisme privé auprès du CES : connaissant la démarche et les objectifs de vos enquêtes SYPS, serait-il possible d'effectuer une étude identique, mais dont l'échantillon se composerait de tous les élèves des écoles d'une région écossaise.

L'échantillonnage des enquêtes SYPS

Pour ce qui concerne l'échantillon des études LONGITUDINALES, le CES a opté pour la technique de l'échantillonnage aléatoire simple⁵. En effet, l'échantillonnage se fait sur base de la date de naissance. Les élèves dont le jour de la date de naissance tombe, par exemple, les 8, 18 et 28 de chaque mois sont repris dans l'échantillon. Cette façon de procéder permet de cerner environ 10% de la population.

Il est à remarquer que le choix de la date de naissance comme critère d'échantillonnage a pour conséquence de biaiser la représentativité des écoles : une école peut n'avoir que 3 ou 4 représentants alors qu'une autre peut en avoir 20.

⁵ Cette méthode consiste à extraire au hasard dans la population les éléments qui composent l'échantillon. Pour plus de précisions, cfr chap 3, section 3.2.

L'administration d'une enquête et la récolte des données

L'administration d'une enquête et la récolte des données sont deux étapes importantes, mais astreignantes du processus d'enquête.

Après analyse détaillée de ces deux étapes, on a pu observer qu'elles se composent de diverses tâches. Ces tâches ont été réparties entre deux départements : le département ADMINISTRATIF et le département ENCODAGE. Nous vous présentons ce schéma qui permet de distinguer le travail effectué par chacun de ces deux départements :

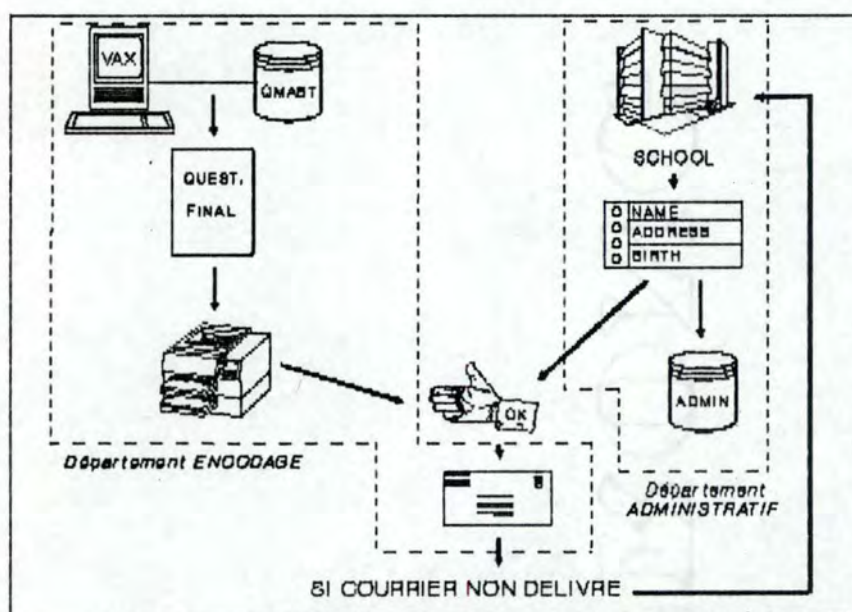


Fig. 2.1.2 : Schéma des opérations "administratives"

Le département ENCODAGE se charge des opérations nécessaires à la production et à l'envoi de tous les questionnaires, à savoir :

1. extraire de QUESTMAST⁶ le questionnaire définitif,

⁶ QUESTMAST est le progiciel d'aide à la conception de questionnaires.

2. préparer les écrans de saisie destinés à l'encodage,
3. produire les copies,
4. imprimer les adresses sur étiquettes auto-collantes,
5. envoyer les questionnaires (coller les étiquettes et timbrer les enveloppes),
6. distribuer des cartes de rappel.

Il va sans dire que pour réaliser les opérations 1 et 2, le département ENCODAGE recourt à l'utilisation de programmes informatiques alors que l'opération 5 s'effectue encore manuellement.

L'objectif de l'opération 6 est d'augmenter le taux de renvoi des questionnaires. Le déroulement de cette opération est le suivant :

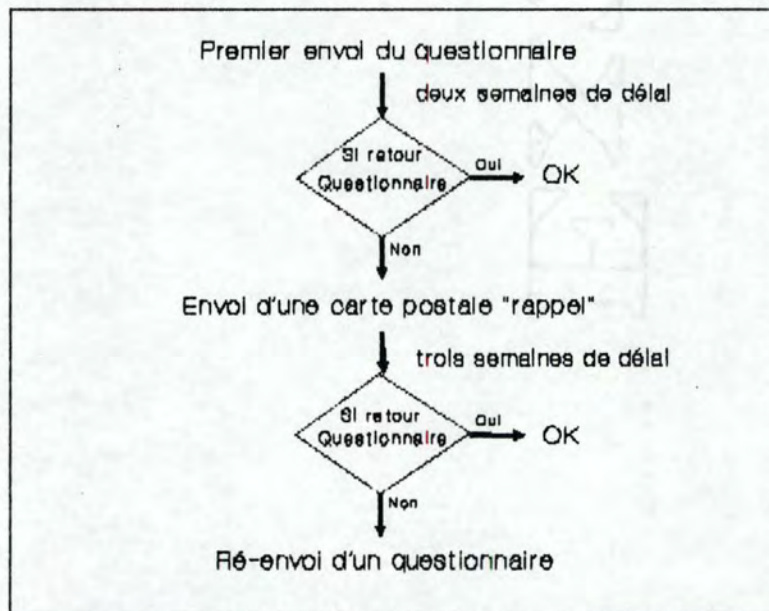


Fig. 2.1.3 : Schéma du mécanisme des cartes "rappel"

Le département ADMINISTRATIF se charge, quant à lui, des tâches purement administratives telles que :

1. se procurer les adresses des individus,

2. mettre à jour la base de données ADMIN contenant les informations relatives à ces mêmes individus,
3. déceler les individus "illégitimes",
4. retrouver traces des individus manquant à l'appel,
5. relancer les "réticents".

Pour mener à bien les deux premières opérations, le département ADMINISTRATIF recourt aux informations que possède le Scottish Education Department (SED) dont dépendent toutes les écoles d'Ecosse.

L'opération 4 combinée au fait qu'un questionnaire peut revenir avec la mention "COURRIER NON DELIVRE" peut conduire le département à réaliser de nouvelles opérations : contacter à nouveau le SED, s'adresser à d'autres organismes afin d'obtenir une adresse plus récente.

Il nous faut avant de conclure faire certaines remarques. La méthode de travail du CES n'est nullement statique, du fait même de l'objet des études que ce centre réalise ; en effet, les individus interrogés s'inscrivent dans un environnement soumis à de multiples changements tels que les réformes des programmes d'enseignement, les créations de programmes de formation, ... Il semble évident dès lors que les enquêtes doivent évoluer de manière à intégrer ces différentes modifications.

De même, sur les 20 années considérées, diverses mutations technologiques se sont produites entre autres, dans le monde informatique. En désirant recourir aux disponibilités nouvelles qu'offre l'évolution technologique, le CES modifie constamment sa méthode de travail.

On comprend dès lors comment pendant ce laps de temps, le centre a pu accumuler un ensemble complexe et interdépendant de bases de données. C'est pourquoi il a besoin d'outils pour saisir et manipuler ces bases de données documentation et pour les aider à les structurer.

2.2 Les logiciels générateurs de questionnaires

2.2.1 Le traitement d'enquêtes par le système BLAISE⁷

BLAISE est le nom d'un système dont l'objectif est d'améliorer la collecte de données et de faciliter l'élaboration d'une enquête. Ce système a été développé au "Netherlands Central Bureau of Statistics" en version PC.

Pour BLAISE, le questionnaire joue un rôle clé dans la production d'informations nécessaires à une gestion efficace des bases de données. Pour obtenir une 'complétude' et une cohérence parfaites, le système BLAISE traite toutes les informations que l'on trouve dans les différentes phases de l'élaboration d'une enquête : la conception d'un questionnaire, la collecte de données, l'édition des données et, enfin, l'analyse de ces données.

Dans la première phase de l'élaboration d'une enquête, à savoir la conception du questionnaire, des éléments essentiels tels que les questions, les réponses possibles et les relations existant entre ces réponses, doivent être définis. La spécification de ces éléments est introduite par un éditeur de texte et est ensuite stockée dans un fichier. Des programmes dont le but est alors d'extraire les informations pertinentes de ce fichier, sont exécutés et permettent ainsi une gestion aisée des données.

Le système BLAISE produit trois types de programmes de saisie et de codage des données : CADI (Computer Aided Data Entry), CAPI (Computer Aided Personal Interviewing) et CATI (Computer Aided Telephone Interviewing). Notre propos n'étant pas de donner une description complète de ce système, nous ne dresserons pas ici la liste des objectifs et mécanismes de ces trois programmes. Toutefois, signalons que CADI est conçu pour la réalisation d'enquêtes postales traditionnelles, CAPI pour les enquêtes par interviews "face-à-face" et CATI pour les enquêtes par interviews téléphoniques.

⁷ Inspiré de "Integrated Survey Processing with BLAISE system" de J. BETHLEHEM, A. HUNDEPOOL, Netherlands Central Bureau of Statistics.

Grâce à une configuration adaptée, le système BLAISE est compatible avec des systèmes statistiques tels que SPSS et STATA ainsi qu'avec des systèmes de gestion de bases de données comme PARADOX.

2.2.2 Le système SNAP⁸

Le logiciel SNAP2 a été créé en 1988 par Mercator Computer Systems. Il existe en version PC et se compose de deux programmes : CATILOG et SNAP.

CATILOG est un système interactif d'interviews. Il permet la conception du questionnaire et l'impression de celui-ci. Le caractère interactif offre un encodage direct et une assistance permanente, c'est-à-dire que CATILOG gère automatiquement les sauts des questions filtres⁹. Par exemple, supposons qu'à l'écran, nous ayons la question suivante :

Q1. Etes-vous membre de ce gymnase ?

OUI

☐

NON

☐

Si votre réponse est NON, passez à la question 3

Dans ce cas, CATILOG affichera directement la question 2 si la réponse à Q1 est OUI, et la question 3 dans la cas inverse.

Une connection directe est établie entre les deux programmes CATILOG et SNAP.

Le programme SNAP s'occupe de l'analyse des données. Il met à la disposition de l'utilisateur des outils qui permettent le calcul de certains paramètres et la comparaison entre variables.

En se référant aux trois types de programmes générés par BLAISE, SNAP2 se situerait plutôt entre CAPI et CATI.

8 Inspiré de "SNAP : Survey Analysis Package" de MERCATOR COMPUTER SYSTEMS.

9 Ce type de questions sera développé au chapitre 3, section 3.1.3.

2.2.3 Le système QUESTMAST¹⁰

Les programmeurs du CES ont développé QUESTMAST qui a été mis en service en 1981.

QUESTMAST est un progiciel conçu pour aider les chercheurs, désireux de mettre sur pied une enquête par questionnaire, dans le développement et la production de questionnaires. QUESTMAST offre, entre autres, des facilités pour concevoir le questionnaire, gérer la saisie des données et établir des relations entre le questionnaire et le fichier de données-résultat. Il génère automatiquement les structures et les relations propres à la création d'une base de données. QUESTMAST a été créé pour être compatible avec des applications statistiques tel que, par exemple, SPSS.

Si on reprend les programmes intégrés dans le système BLAISE, on remarque que QUESTMAST correspond au premier des trois programmes, CADI. En effet, QUESTMAST n'est conçu que pour la réalisation d'enquêtes postales traditionnelles.

Actuellement, une version pour PC de QUESTMAST est en cours de conception au CES.

En effet, QUESTMAST doit être mis à jour pour des raisons diverses :

- la première raison est que QUESTMAST est dépendant du Edinburgh Multi-Acces System (EMAS) qui sera hors service en 1992 ;
- la deuxième, l'interface est dépassé par rapport aux exigences actuelles ;
- la troisième est que QUESTMAST travaille sur un seul fichier alors que les enquêtes longitudinales impliquent que le logiciel utilisé puisse gérer plusieurs fichiers contenant chacun un questionnaire.

10 Inspiré de "QUESTMAST : a package to aid the design and construction of questionnaires" de J.M LAMB, 1981.

2.3 L'élaboration d'un système de documentation

Etant donné l'ampleur des informations disponibles actuellement dans les bases de données, il n'est pas surprenant d'observer des utilisateurs réticents à l'idée de consulter ces bases, véritables montagnes d'informations.

Ne pouvant pas toujours connaître les différents domaines couverts par ces données et, pire encore, ne sachant dans quelles bases chercher ces données, l'utilisateur se voit obligé de recourir aux mêmes bases de données à plusieurs reprises.

Dès lors, un système qui permette une consultation aisée et efficace de ces bases serait le bienvenu. Ce système est la base de données documentation.

2.3.1 Qu'est-ce qu'une base de données documentation ?

Une base de données documentation consiste en un dictionnaire de données bien spécifiques dont le rôle est de documenter les données contenues dans d'autres bases. Ce dictionnaire se présente également sous la forme d'une base de données et doit pouvoir être manipulé comme tel.

L'implémentation d'une telle base de données permettra aux utilisateurs (chercheurs, analystes, archivistes) de n'avoir à interroger qu'une seule source pour obtenir un maximum d'informations sur les données recherchées.

2.3.2 Les objectifs d'une base de données documentation

Un tel système de documentation a deux objectifs :

- 1) fournir une documentation exhaustive de tous les éléments des bases de données afin de faciliter la recherche de ceux-ci ;
- 2) à terme, offrir aux utilisateurs une interface d'interrogation "on-line".

Dans le cas particulier des enquêtes par questionnaire, quelles sont les contraintes dont devra tenir compte l'architecture de la base de données documentation ?

- 1) Les étapes du processus : la première solution réside en la récolte des informations en simultanéité avec le déroulement du processus d'enquête (inconvenient : la monopolisation des ressources pour l'extraction et le stockage des éléments). La seconde consiste à récolter ces informations après le déroulement du processus.
- 2) Les connaissances personnalisées : dans bien des cas, les concepteurs des enquêtes détiennent des informations pertinentes concernant l'objet de l'enquête ou les conditions de sa réalisation. Au fur et à mesure que le temps passe, ces éléments sont perdus.

2.3.3 Le prototype d'une base de données documentation : DOCDB

DOCDB est un projet appartenant au département informatique du CES et qui est financé par le ESRC¹¹ au même titre que les projets de recherche déposés par le département recherche.

Ce projet a pour objectif de documenter toutes les données relatives à un processus d'enquête. Il est actuellement en cours de réalisation et c'est dans le cadre de ce projet que nous avons travaillé lors de notre stage¹².

Les programmeurs du CES ont décidé de procéder en deux phases pour aboutir à la réalisation complète de DOCDB : la phase de **REPLISSAGE** et la phase de **CONSULTATION**. Ces phases sont illustrées dans la figure suivante :

11 Rappelons que ESRC est l'abréviation de Economic and Social Research Council.

12 La description de notre travail de stage sera l'objet du chapitre 6.

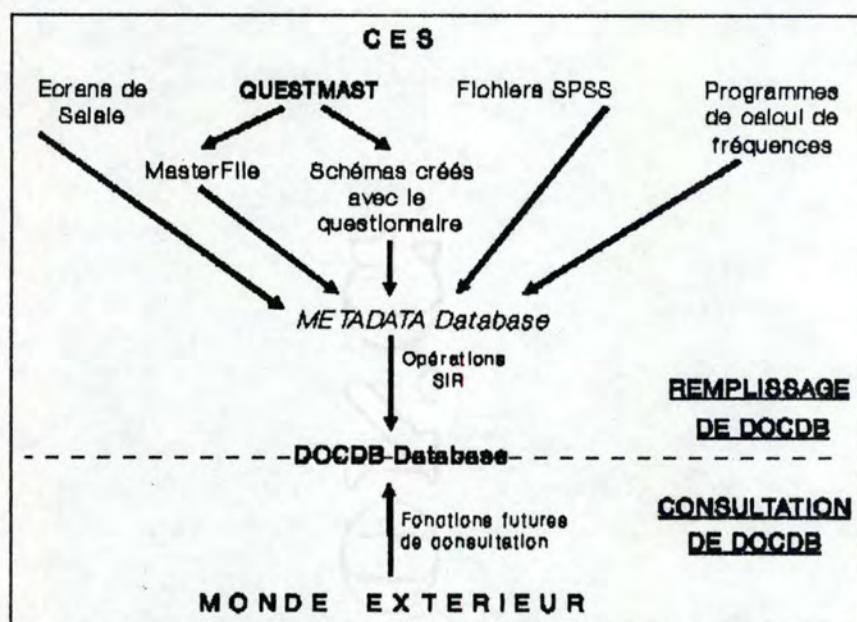


Fig. 2.3.1: Le projet DOADB (en cours de réalisation)

La première de ces deux phases se décompose encore en deux étapes :

- * l'étape de CREATION d'une base de données intermédiaire, appelée METADATA dont le but est de rassembler, de cataloguer les éléments provenant de diverses sources et nécessaires à la documentation,
- * l'étape de TRANSFERT des informations contenues dans METADATA vers la base documentation DOADB.

N'ayant pas encore déterminé de manière complète et précise les éléments pertinents du processus d'enquête qui doivent obligatoirement se trouver dans DOADB pour permettre une consultation aisée des bases de données, les programmeurs du CES n'ont, jusqu'à présent, conçu qu'un schéma théorique et provisoire de DOADB.

Les schémas Entité/Association que nous avons défini et décrit au chapitre 5, doivent être considérés comme une proposition à la conception du schéma de DOCDB. D'ailleurs, ces schémas ont eu pour conséquence de solutionner certains problèmes de redondance et de relation entre entités qui résidaient dans le schéma théorique existant de DOCDB.

C'est une raison pratique qui justifie l'emploi d'une base de données intermédiaire. En effet, on se trouve face à une multitude d'informations qui proviennent de sources très diverses qu'il faut structurer pour rendre efficace l'utilisation de DOCDB. Se basant sur la structure théorique de DOCDB, il est possible de sélectionner les éléments de ces sources qui seront utiles à la documentation et stockés provisoirement dans METADATA.

DOCDB adoptera une structure définitive lorsque l'on déterminera les fonctions de consultation requises par les chercheurs. La structure établie permettra aussi de déterminer les accès nécessaires à la base documentation. Mais c'est là que réside le problème, car les chercheurs et les personnes autorisées à consulter les bases de données n'ont pas encore défini leurs besoins en terme de fonctions de consultation. Plus ces fonctions seront déterminées de manière précise, plus le transfert des informations de METADATA vers DOCDB s'effectuera de manière optimale.

La conception des fonctions de consultation sera basée sur l'utilisation de concepts-clés tels que la QUESTION.

En effet, la question est le point de départ de la consultation. A partir des questionnaires distribués par le CES, les personnes désireuses de consulter les bases de données du centre, sélectionneront les questions qui sont susceptibles de les intéresser pour leur propre recherche. La base de données documentation devra alors renseigner la personne non seulement sur les résultats mêmes de la question dans l'enquête, mais aussi sur le contexte de cette question.

La conception des fonctions de consultation et le transfert des informations feront l'objet des perspectives futures

d'implémentation et pour faciliter celle-ci, les programmeurs ont introduit la notion de raisonnement déductif¹³ au niveau de la question.

Mais la tâche n'est pas mince

13 L'application de cette notion au mécanisme de la question filtre est reprise en Annexe 6.

3. CONSIDERATIONS THEORIQUES

3.1 Les phases générales d'une enquête par questionnaire

3.1.1 Détermination des objectifs et évaluation des moyens d'une enquête

En tentant de délimiter l'objet de l'enquête, on arrive à déterminer l'étendue de l'enquête et une première estimation des moyens à mettre en oeuvre.

Définition de l'objet de l'enquête

L'objet de l'enquête est un problème à étudier. En général, le problème posé est ensuite traduit sous forme d'hypothèses, qui seront alors testées sur le terrain et auxquelles les chercheurs tenteront d'apporter une évaluation.

Recherche préalable

Ayant défini l'objet de l'enquête, les chercheurs consultent toute la documentation disponible sur des recherches déjà menées à ce propos ou dans ce domaine. Ils examinent aussi les différentes méthodes qui ont été utilisées, tout ceci, afin d'éviter d'effectuer une enquête sur un objet déjà étudié.

Inventaire des moyens

En principe, cet inventaire reprend les problèmes liés à l'établissement d'un budget, du "calendrier" de l'enquête, des conditions générales (par exemple, des autorisations à faire valoir, ...). Mais tous ces problèmes sont dictés irrémédiablement par la contrainte financière. Celle-ci détermine d'ailleurs d'autres aspects :

- 1) le personnel nécessaire (non seulement au niveau quantitatif, mais aussi qualitatif),

- 2) les sources documentaires nécessaires,
- 3) les besoins matériels (par exemple, les moyens d'impression).

3.1.2 Définition de l'échantillon

Le tout n'est pas d'avoir déterminé l'objet de l'enquête, il faut encore savoir qui interroger.

On appelle population la totalité ou l'ensemble des éléments soumis à l'investigation et sur lesquels on désire obtenir certaines informations. Il est souvent difficile, voire même impossible, d'examiner toute la population. En conséquence, on considère seulement une partie, appelée échantillon.

Dès lors, dans la plupart des études réalisées, c'est l'examen de l'échantillon qui permet de tirer des conclusions et propriétés sur l'ensemble tout entier de la population. Pourquoi utiliser un échantillon, c'est-à-dire un nombre limité d'observations, plutôt qu'un recensement complet, a priori plus fiable ? En voici quelques raisons :

- 1) d'ordre économique : on engage moins de ressources pour cerner un échantillon que pour établir un recensement complet ;
- 2) d'ordre pratique : une population est partiellement inaccessible ;
- 3) d'efficacité statistique : il est plus facile de contrôler l'exactitude des erreurs et des transmissions de données sur un plus petit nombre d'observations.

Il s'agit donc de créer un échantillon qui reprenne l'ensemble des caractéristiques de la population dans une juste proportion et sans générer de biais. La constitution d'un tel échantillon pose un double problème :

- 1) quelle doit être la taille de l'échantillon ?
- 2) comment déterminer les individus qui devront faire partie de cet échantillon ?

Ces deux aspects sont fondamentaux pour l'élaboration d'une enquête et nous conseillons au lecteur de se référer au chapitre traitant de l'échantillonnage¹.

3.1.3 Rédaction du questionnaire

La réalisation d'un questionnaire est toujours un moment délicat. Le travail du chercheur consiste, après multiples changements au niveau du choix des questions à poser, en la réalisation d'un projet de questionnaire simple, clair, sobre, répondant aux objectifs qui ont été fixés. A la fin de cette phase, les chercheurs disposent d'un ensemble de questions. Il faut encore déterminer quelle forme prendra chacune de ces questions.

Les types de questions

Dans cette partie plus spécialement, mais aussi par la suite, nous nous limiterons aux questions INTERROGATIVES² qui sont généralement de quatre types.

Le premier type est la question fermée dont les réponses sont fixées par le concepteur du questionnaire. Par conséquent, la personne interrogée est dans l'obligation de choisir parmi ces seules réponses qui lui sont proposées.

Exemple :

Question 1 : Etes-vous, actuellement, inscrits dans programme de formation ?

.....	OUI	O
.....	NON	O

Le deuxième type de questions se base sur le principe de la question fermée. En effet, les questions filtres sont :

1 Cfr Chapitre 3, section 3.2.

2 Nous avons choisi de traiter ce type de questions car c'est le type le plus fréquemment utilisé. Le lecteur qui désire obtenir un relevé exhaustif des autres types de questions et de leur description, peut se référer à l'ouvrage de Cl. Javeau "L'enquête par questionnaire".

- 1) soit des questions qui introduisent les questions suivantes ;
- 2) soit des questions contrôles qui servent à vérifier, par recoupement, le contenu d'une réponse à une question se situant à un autre endroit du questionnaire.

Mais, dans les deux cas, la première question de cette cascade de questions est une question fermée.

Exemple :

Question 2 : Avez-vous quitté l'école ?

..... OUI 0
 NON 0

Question 3 : SI OUI, pensez-vous que le fait d'avoir quitté l'école était la bonne décision ?

..... OUI 0
 NON 0

Question 4 : SI NON, Avez-vous déjà commencé un programme en vue de l'obtention d'un certificat national ?

..... OUI 0
 NON 0

Le troisième type est la question ouverte. Dans ce cas, le répondant a la liberté de répondre ce qu'il désire, même si parfois, la réponse est canalisée. La différence principale par rapport à la question fermée est l'impossibilité de prévoir l'attitude du répondant.

Exemple :

Question 5 : Citez les différentes marques de voiture que vous avez achetées :

1.
 2.
 3.
 4.
 5.

Son utilisation est recommandée quand :

- 1) il est impossible de prévoir les différentes possibilités de réponses ;
- 2) il y a un trop grand nombre de catégories pour utiliser une question fermée (exemple : l'ensemble des marques de voiture).

La tendance actuelle consiste à réduire le nombre des questions ouvertes pour obtenir une réponse plus précise ou pour permettre un dépouillement plus rapide.

Pour atteindre ces objectifs, il est parfois utile de transformer une question ouverte en une question fermée. Voici un exemple :

On a la question suivante :

Question 6 : En quelle année, votre maison a-t'elle été construite :

- - - -

que l'on transforme de la manière suivante :

Question 6 : En quelle année, votre maison a-t'elle été construite ?

.....	Avant 1900	O
.....	Entre 1900 et 1919	O
.....	Entre 1920 et 1939	O
.....	Entre 1940 et 1959	O
.....	Entre 1960 et 1979	O
.....	Après 1979	O

Le quatrième type de questions est la question semi-fermée (ou semi-ouverte). Ce type cumule les avantages et les inconvénients des premier et troisième types : le concepteur a la possibilité de spécifier la plupart des réponses, mais peut laisser la capacité au répondant de fournir une réponse.

Exemple :

Question 7 : Etes-vous **MAINTENANT** :

- | | | |
|-------|---|---|
| | inscrit à un programme de formation pour jeunes ? | O |
| | travailleur à temps plein ? | O |
| | au chômage et demandeur d'emploi ? | O |
| | inscrit à des cours du jour dans une école ? | O |
| | inscrit à des cours du jour dans un établissement supérieur ? | O |
| | occupé à exercer une autre activité | |
| | Quel type d'activité ? | |
| | | |

Après la description des différents types de questions interrogatives, nous vous proposons à présent de prendre connaissance des règles nécessaires à la bonne rédaction des questions.

La rédaction des questions

Une question bien construite devra :

- * être précise : il faut vérifier que la question ne soit pas à double sens,
- * ne contenir qu'une seule idée,
- * être courte : il faut éviter l'utilisation de négation et de double négation,
- * éviter la personnalisation directe, surtout lorsque le thème abordé est délicat,
- * être comprise par les personnes interrogées (là réside tout le problème de la sémantique),
- * éviter l'utilisation de mots qui ont une signification plus ou moins péjorative ou affective,
- * être objective.

Après que les questions aient été préparées et libellées une à une, le questionnaire est prêt à être construit.

La construction du questionnaire

La construction d'un questionnaire ne peut pas être négligée. Une logique d'enquête veut, par exemple, que l'on débute par une ou plusieurs questions d'ordre général destinées à en introduire les objectifs.

Le questionnaire doit se dérouler de façon à ne pas provoquer la lassitude du répondant.

Chaque sujet doit être entièrement cerné dans une séquence de questions. Il faut éviter des "aller-retour" qui énervent le répondant.

Des commentaires en italique doivent être insérés pour expliquer comment répondre à telle ou telle question ou pour signaler le passage d'un thème à un autre.

Il est conseillé de marquer de manière claire la question vers laquelle on est reporté dans le cas de questions filtres.

Le questionnaire ne doit pas être trop long. En effet, il a été démontré que le pourcentage de non-réponse est proportionnel à la longueur du questionnaire.

Le problème matériel de mise en page a toute son importance, que celle-ci soit effectuée manuellement ou au moyen d'un logiciel :

- * ne pas négliger le choix des caractères et du papier,
- * ne jamais scinder une question d'une page à l'autre,
- * soigner particulièrement l'espacement des questions et la présentation de celles-ci.

L'introduction du questionnaire est une opération indispensable. Une lettre, accompagnant le questionnaire, doit expliquer brièvement le "pourquoi" de l'enquête. Il est aussi conseillé de faciliter l'envoi de retour grâce à une enveloppe de réponse timbrée et adressée au destinataire, ceci dans le cas d'une enquête postale.

Comme il a déjà été dit précédemment, la construction d'un questionnaire peut être assistée par des moyens informatiques. Il existe des logiciels qui permettent un choix diversifié de

formes pour les questions, un positionnement automatique sur la page, ...

Cette phase n'est pas la seule qui recourt à l'informatique. En effet, on fera encore appel à celle-ci dans la phase de codage et d'encodage des données.

3.1.4 Dépouillement et codage des données

Le codage est une opération qui consiste à donner à chaque catégorie de réponses un numéro de code, alors que la codification consiste à faire entrer chaque réponse dans une des catégories de code. En pratique, la codification a donc pour objectif de transformer une réponse concrète en une réponse codée.

Il est dès lors logique de constater que les questions de type fermée remportent un vif succès auprès des codeurs. Par contre, pour les questions ouvertes, le codeur se voit confronté au problème d'interprétation des réponses.

Lorsque les questions ont été codées, il faut les dépouiller, c'est-à-dire qu'il faut en dégager les résultats intéressants par rapport aux hypothèses définies lors de la première phase.

Le dépouillement consiste principalement en l'opération de COMPTAGE : c'est le décompte pur et simple des réponses pour une question déterminée.

Exemple : Croyez-vous que le port de la ceinture de sécurité est un élément essentiel à la sécurité routière ?

.....	OUI	O
.....	NON	O
.....	JE NE SAIS PAS	O

On fera le décompte des "oui", des "non" et des "je ne sais pas", que l'on conservera tel quel (fréquence absolue) ou que l'on ramènera à 100 (fréquence relative).

Dans cette quatrième phase, on trouvera encore les opérations de vérification de cohérence des données encodées

et de contrôle de la représentativité de l'échantillon. La réalisation de l'enquête à partir d'un échantillon se traduit par un certain nombre de non-réponses et par des questionnaires incomplets. Il est donc nécessaire de vérifier que la représentation des réponses est conforme à celle de la population étudiée. Si la différence est petite, elle sera intégrée dans la marge d'erreur. Si la différence est grande, il faudra recourir à une pondération des résultats³.

Il est à remarquer que, dans certains ouvrages, les opérations décrites dans cette quatrième phase sont reprises sous les termes génériques de "traitement des données".

3.1.5 Analyse et interprétation des résultats

A présent que la réalisation matérielle est terminée et que tous les tableaux de résultats ont été constitués, les chercheurs disposent des données nécessaires pour tester les hypothèses de départ.

L'analyse des résultats consiste en l'évaluation des résultats obtenus par rapport aux hypothèses posées. Ces hypothèses sont alors précisées, mesurées, confirmées ou infirmées. Bien entendu, le calcul statistique est nécessaire pour évaluer les "mouvements" des différentes variables que l'on étudie dans les résultats. Il faut donc calculer la part, l'importance et l'influence statistique de chacune de ces variables⁴.

Lorsque toutes ces variables ont été interprétées, il reste encore aux chercheurs à rédiger le rapport de l'enquête. Ce rapport doit avant tout comporter :

- * la définition des objectifs de l'enquête et la détermination de l'échantillon,
- * la présentation des résultats, de tableaux clairs et de graphiques significatifs,
- * la méthodologie utilisée.

3 Notre propos n'est pas ici de détailler les différentes méthodes de pondération.

4 Pour ce faire, on fait appel à des méthodes telles que les régressions, les calculs de corrélation, ...

En conclusion, on peut affirmer que, si la rédaction du rapport est la touche finale du processus d'enquête, celle-ci se prépare tout au long de l'enquête et s'inspire de beaucoup d'éléments qui interviennent dans les différentes phases de l'enquête.

Dans le chapitre 5, nous tenterons de modéliser ces différentes phases et de développer celles qui sont a priori les plus faciles à informatiser.

Mais, au préalable, nous vous proposons de prendre connaissance des méthodes d'échantillonnage.

3.2 Les méthodes d'échantillonnage

3.2.1 Les notions⁵

L'échantillonnage est une opération qui consiste à prélever un certain nombre d'éléments (échantillon), dans l'ensemble des éléments que l'on veut observer ou traiter (population).

L'échantillon est donc un ensemble d'éléments à propos desquels on a effectivement recueilli des données. Alors que la population est l'ensemble des éléments qui possèdent les caractéristiques qu'on veut observer.

Il est souvent difficile, voire impossible, d'examiner toute la population. C'est l'examen de l'échantillon qui doit permettre de tirer des conclusions sur l'ensemble de la population : il s'agit du raisonnement inductif. De là, la nécessité de choisir un échantillon représentatif, miroir de la population. Un échantillon est dit représentatif d'une population pour un caractère si il n'y a aucune raison de penser que la valeur de ce caractère puisse différer dans l'échantillon et dans la population.

Pour qu'une méthode d'échantillonnage conduise à un échantillon représentatif, il faut d'une part, en déterminer

5 Définitions extraites de "Concepts et Méthodes de la Statistique" de L. D'HAINAUT.

la taille et d'autre part, que tous les individus de la population aient la même chance d'être sélectionnés. Le problème de la taille de l'échantillon sera traité dans la section 3.2.2. La description des différentes méthodes d'échantillonnage fera l'objet de la section 3.2.3.

3.2.2 La taille de l'échantillon

Puisqu'en pratique, il n'est pas toujours possible et il est presque toujours plus onéreux d'effectuer toutes les mesures nécessaires pour avoir une information complète, on se limitera le plus souvent à recueillir des données sur un petit nombre d'éléments et on en déduira des propriétés sur l'ensemble tout entier. Comment déterminer ce nombre d'éléments ?

Soit X une variable aléatoire de densité de probabilité $f_X(x)$. Supposons que l'on tire un échantillon de deux valeurs de X , disons x_1 et x_2 , au hasard. Le couple (x_1, x_2) détermine un point du plan et l'ensemble de ces couples définit une population à 2 dimensions.

On souhaite décrire la distribution de cette population. La réponse est simple.

Puisque l'échantillonnage est aléatoire, la valeur de la première observation n'influence pas la valeur de la seconde, et par conséquent, les deux variables aléatoires X_1 et X_2 sont indépendantes au sens des probabilités. La densité de (X_1, X_2) est donc le produit des densités marginales.

$$f_{X_1, X_2}(x_1, x_2) = f_{X_1}(x_1) * f_{X_2}(x_2)$$

$$\Pr(X_1=x_1 \text{ et } X_2=x_2) = \Pr(X_1=x_1) * \Pr(X_2=x_2)$$

Si l'on généralise à un échantillon aléatoire de n observations, issues d'une population dont la densité de probabilité est $f(x)$, on obtient :

$$f_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) = f_{X_1}(x_1) * f_{X_2}(x_2) * \dots * f_{X_n}(x_n)$$

Il ne faut pas oublier que ce résultat n'est valable que si l'échantillonnage est aléatoire.

Une statistique est une fonction des observations de l'échantillon. Plusieurs mesures sont couramment utilisées pour résumer l'ensemble des données d'un échantillon ou d'un recensement. Les plus connues sont notamment :

* la moyenne d'échantillonnage

la moyenne est un paramètre de valeur centrale, égal à la somme de toutes les valeurs observées divisée par l'effectif :

$$\overline{X} = \frac{\sum_{i=1}^n x_i}{n}$$

où n est la taille de l'échantillon

* la variance d'échantillonnage

la variance mesure le degré de dispersion des valeurs observées autour de la moyenne en terme de carrés des écarts :

$$S^2 = \frac{\sum_{i=1}^n (x_i - \overline{X})^2}{n}$$

* l'écart-type ou déviation standard

comme la variance est exprimée en unités représentant le carré des unités des valeurs originales, afin d'obtenir une mesure de dispersion exprimée dans la même unité que celle des valeurs originales, on définit l'écart type :

$$S = \sqrt{S^2}$$

En fonction de l'échantillon choisi, une statistique donnée prend différentes valeurs. L'ensemble des valeurs possibles et leur probabilité, représente la distribution d'échantillonnage de la statistique. La valeur particulière que prend la

statistique après observations n'est donc qu'une valeur parmi un ensemble de possibles.

La moyenne d'échantillonnage \bar{X} tend à se rapprocher de la moyenne ou espérance μ de la population, quelle que soit la distribution de la population.

Soit X_1, X_2, \dots, X_n un échantillon aléatoire de taille n d'une population de densité $f(x)$,
alors $E(\bar{X}) = \mu$

Autrement dit, si on répète plusieurs fois un processus d'échantillonnage et que l'on calcule le \bar{X} associé, les valeurs trouvées auront tendance à s'agglomérer autour de l'espérance ou moyenne de la population μ , la dispersion étant d'autant plus petite que le nombre d'observations est grand, comme l'indique le théorème suivant :

Soit X_1, X_2, \dots, X_n un échantillon aléatoire de taille n d'une population de densité $f(x)$,
d'espérance μ et de variance σ^2 ,
alors $\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$

Ainsi, plus l'échantillon est grand, plus la moyenne d'échantillonnage se rapproche de la moyenne de la population μ en ce sens que sa variance diminue.

Ces différents théorèmes établis, la question demeure : "Combien doit-on prendre d'individus pour que l'échantillon soit représentatif ?". En effet, ces théorèmes ne nous permettent de tirer que la conclusion suivante : un nombre suffisamment grand.

Une autre mesure de la façon dont la moyenne de l'échantillon se rapproche de l'espérance de la population s'avère nécessaire.

Par le Théorème Central Limite, nous savons :

Soit $f(x)$ une densité de probabilité de moyenne μ et de variance finie σ^2 .

Soit \bar{X}_n la moyenne d'un échantillon aléatoire de taille n issu de $f(x)$.

Soit Y_n la variable aléatoire

$$Y_n = \frac{\bar{X}_n - \mu}{\sigma} * \sqrt{n}$$

La loi de probabilité de Y_n tend vers la loi Normale de moyenne 0 et de variance 1 (normale réduite), quand n tend vers l'infini. Y_n est distribué selon une Normale Réduite.

Ou encore, la variable aléatoire \bar{X}_n tend vers une loi normale d'espérance μ et de variance $\frac{\sigma^2}{n}$

$$\bar{X}_n \text{ est distribué selon une Normale } \left(\mu, \frac{\sigma^2}{n} \right)$$

Sachant que Y_n suit une loi Normale Réduite,

$$\bar{X}_n - \frac{\sigma}{\sqrt{n}} * Q_{NR}\left(1 - \frac{\alpha}{2}\right) \leq \mu \leq \bar{X}_n + \frac{\sigma}{\sqrt{n}} * Q_{NR}\left(1 - \frac{\alpha}{2}\right)$$

où la valeur de $Q_{NR}\left(1 - \frac{\alpha}{2}\right)$ est lue dans la table de la loi Normale Réduite
et

où α est le risque d'erreur.

3.2.3 Les méthodes d'échantillonnage

Différentes méthodes d'échantillonnage seront développées parmi lesquelles l'échantillonnage aléatoire simple, stratifié et en grappes.

Modèle fondamental : l'échantillonnage aléatoire simple

Par échantillonnage aléatoire simple, toute unité de la population doit avoir la même chance de figurer dans l'échantillon. Pour assurer la même chance à tous les individus d'être sélectionnés, le procédé le plus adéquat est le tirage au sort. Notons toutefois que dans le cas où la population est trop grande, on peut recourir aux tables des nombres aléatoires.

Exemple : Soit un chercheur désirant réaliser une enquête sur les entraîneurs des équipes de football en Belgique. A chaque nom d'entraîneur est associé un nombre de 1 à 157. Chaque nombre est inscrit sur un papier qui sera placé dans une urne.

Après avoir mélangé, on tire 35 papiers ; les noms correspondant constitueront l'échantillon du chercheur.

L'exemple ci-dessus représente l'échantillonnage aléatoire simple ; il existe d'autres méthodes d'échantillonnage aléatoire entre autres l'échantillonnage systématique, obtenu en choisissant un individu au hasard, puis les suivants à intervalle régulier de manière à obtenir le nombre désiré.

L'échantillonnage stratifié

La méthode d'échantillonnage aléatoire simple peut être améliorée dans le cas d'une population de référence hétérogène. La population est alors partagée en sous-groupes, appelés strates, qui diffèrent l'une de l'autre par rapport au caractère observé. Par la suite, on prélève dans chaque strate un échantillon selon un processus aléatoire.

Une des raisons au recours à ce type d'échantillonnage, est que celui-ci permet de mettre en évidence un échantillon plus efficace que celui obtenu suite à un processus aléatoire.

Exemple : Si on fait un sondage électoral, on peut diviser a priori la population en catégories socio-professionnelles, car il semble évident que, dans l'ensemble, les catégories défavorisées ne votent pas de la même façon que les catégories les mieux nanties.

Les strates ainsi élaborées constituent chacune un groupe homogène, mais comparée l'un à l'autre, elles sont très différentes. L'efficacité de l'échantillonnage stratifié dépend de l'homogénéité à l'intérieur de chaque strate.

Une autre raison est l'assurance que l'échantillon ainsi choisi, reflètera mieux la population, sur la base du ou des critère de stratification. Ceci n'est nullement négligeable, car après échantillonnage aléatoire, il aurait pu se faire que l'échantillon créé reprenne un grand nombre d'individus d'une strate et peu ou aucun individu des autres strates. Grâce à cette méthode, la représentativité de l'échantillon peut être accrue.

Une des tâches du chercheur dans le cas d'un échantillonnage par stratification sera de déterminer le ou les critères de découpage de la population en strates. Ce ou ces critères sont des caractéristiques sur les éléments de la population ou d'autres variables dont le choix est dicté par le domaine de recherche retenu. Cette ou ces variables doivent être définies de manière à accroître d'une part, l'homogénéité au sein de chaque strate et d'autre part, l'hétérogénéité entre les strates. Par la suite, pour chaque strate de la population, le chercheur déterminera par processus aléatoire, l'échantillon sur lequel portera effectivement l'étude.

La méthode d'échantillonnage aléatoire stratifié sur une population de N éléments implique que l'on divise la population en différentes strates de N_1, N_2, \dots, N_L unités (avec $N_1 + N_2 + \dots + N_L = N$).

Quand chaque strate est déterminée, dans chacune d'entre elles est élaboré un échantillon. La taille de ces L échantillons est respectivement n_1, n_2, \dots, n_L . La moyenne d'échantillonnage dans ce cas est

$$\bar{X} = \frac{\sum_{h=1}^L n_h * \bar{X}_h}{n}$$

$$\text{où } \bar{X}_h = \frac{\sum_{i=1}^{n_h} x_{ih}}{n_h}$$

et x_{ih} est la valeur obtenue sur la $i^{\text{ème}}$ unité de la strate h .

Dans chaque strate, si la moyenne \bar{X}_h est un estimateur non biaisé, alors \bar{X} est un estimateur non biaisé de la moyenne de la population μ .

$$E(\bar{X}) = E\left(\sum_{h=1}^L W_h * \bar{X}_h\right) \\ = \sum_{h=1}^L W_h * \bar{X}_h$$

$$\text{avec } W_h = \frac{N_h}{N}$$

$$\text{Or } \mu = \frac{\sum_{h=1}^L \sum_{i=1}^{N_h} x_{ih}}{N} \\ = \frac{\sum_{h=1}^L N_h * \bar{X}_h}{N} \\ = \sum_{h=1}^L W_h * \bar{X}_h$$

Quant à la variance de cet estimateur \bar{X} , elle se calcule de la façon suivante

$$\text{Var}(\bar{X}) = \sum_{h=1}^L W_h^2 * \text{Var}(\bar{X}_h)$$

$$\text{où } \text{Var}(\bar{X}_h) = \frac{\sigma_h^2}{n_h} * \frac{N_h - n_h}{N_h}$$

$$\text{Ainsi, } \text{Var}(\bar{X}) = \sum_{h=1}^L W_h^2 * \frac{\sigma_h^2}{n_h} * \left(1 - \frac{n_h}{N_h}\right)$$

L'échantillonnage par grappes

Si, pour un certain nombre de raisons, dont entre autres des raisons financières, on ne peut dresser la liste complète des unités qui doivent être interrogées, ou si on ne parvient pas à interroger l'ensemble des unités retenues dans l'échantillon élaboré par processus aléatoire, parce que celles-ci sont trop éparpillées géographiquement, on dresse la liste des ensembles regroupant ces unités ; ceux-ci sont appelés les grappes.

Certains exemples de grappes sont donnés dans le tableau qui suit :

Unité sondée	Grappes associées
Individus belges	Provinces Arrondissements Communes Régions Communautés
Entreprises	Régions Zonings industriels

Par la suite, par tirage aléatoire, on choisit certaines de ces grappes et on réalise l'enquête sur l'ensemble des individus de cette grappe ou sur un échantillon de celles-ci ; dans ce cas, on parle d'échantillonnage à plusieurs degrés⁶.

Par opposition à la technique de stratification où les strates constituent chacune un groupe homogène, la grappe doit être tout aussi hétérogène que ne l'est la population elle-même.

Souvent la grappe est constituée par une zone géographique⁷ : il s'agit d'un échantillonnage aréolaire. Ce type d'échantillonnage a l'avantage de permettre de concentrer les interviews et donc, de réduire globalement les coûts (réduction des frais de déplacement et augmentation du rendement quotidien de l'enquêteur). Il a néanmoins le désavantage d'engendrer une ressemblance relativement importante des caractéristiques et attitudes des éléments de la grappe. Ce phénomène est appelé l'effet de grappe. On contournera cet effet en veillant à ce que chaque grappe soit toujours constituée d'éléments différents et en choisissant un grand nombre de grappes sondées.

L'échantillonnage par grappes est généralement assimilé aux techniques d'échantillonnage aléatoire du fait du tirage aléatoire des grappes, mais aussi, de l'échantillon sur

6 Ce point sera développé dans la section suivante.

7 Se référer aux exemples de grappes repris dans le tableau ci-dessus.

chacune des grappes dans le cas d'un échantillonnage à plusieurs degrés.

On a N grappes sur la population et chacune de ces grappes contient M éléments.

Soit N grappes sur la population, chacune contenant M éléments

Soit y_{ij} la valeur observée du $j^{\text{ième}}$ élément dans la grappe i ,

et $y_i = \sum_{j=1}^M y_{ij}$, c'est-à-dire la valeur totale des observations réalisées sur la grappe i .

Dans ce cas, $\bar{Y} = \frac{y_i}{M}$ est la moyenne par grappe

$$\bar{\bar{Y}} = \frac{\sum_{i=1}^N y_i}{N * M} = \frac{\bar{Y}}{M} \text{ est la moyenne de la population}$$

Un échantillonnage aléatoire de n grappes qui contiennent chacune M éléments est réalisé parmi les N grappes de la population.

\bar{y} est un estimateur sans biais de $\bar{\bar{Y}}$,

$$E(\bar{y}) = \bar{\bar{Y}}$$

et

$$\begin{aligned} \text{Var}(\bar{y}) &= \frac{1-f}{n} * \frac{(N*M)-1}{M^2*(N-1)} * \sigma^2 * (1+(M-1)*\rho)) \\ &= \frac{1-f}{n*M} * \sigma^2 (1+(M-1)*\rho)) \end{aligned}$$

où $f = \frac{n}{N}$

$$\rho = \frac{E(y_{ij} - \bar{Y}) * (y_{ik} - \bar{Y})}{E(y_{ij} - \bar{Y})^2} \text{ est le coefficient de corrélation entre les grappes.}$$

4. LES ELEMENTS A PRENDRE EN CONSIDERATION POUR L'ELABORATION DES MODELES

4.1 La définition d'une question

Il a été observé que, pour les enquêtes postales, les concepteurs de questionnaires utilisent plus fréquemment des questions fermées, ceci dans le but d'éviter des problèmes d'interprétation.

Prenons, comme exemple, ces trois questions :

1. Avez-vous quitté l'école ?

OUI ☐ 1 NON ☐ 2 → Si non, allez à la question n° 3

Si oui :

2. Pourquoi avez-vous quitté l'école ? (cocher toutes les cases convenant)

..... parce que j'avais un boulot ☐
 parce que j'en avais marre de l'école ☐
 parce que j'avais assez de connaissances ☐

3. Regrettez-vous votre décision ? (cocher UNE case)

oui ☐ 1
 non ☐ 2
 pas sûr ☐ 3

Quels sont les éléments qui caractérisent chacune de ces questions ?

1) le numéro de la question qui permet de l'identifier,

2) la présentation de la question :

- * le texte de la question que l'on trouve sous le nom de QLIGNE dans QUESTMAST,
- * le nombre et le texte des lignes de réponses qui correspondent aux ALIGNÉ de QUESTMAST,
- * le nombre et le texte des lignes d'en-têtes qui sont connues comme HLINE pour QUESTMAST.

Un autre élément à ne pas négliger est l'enchaînement des questions qui varie selon les réponses apportées.

Il est possible de représenter schématiquement ces différentes caractéristiques :

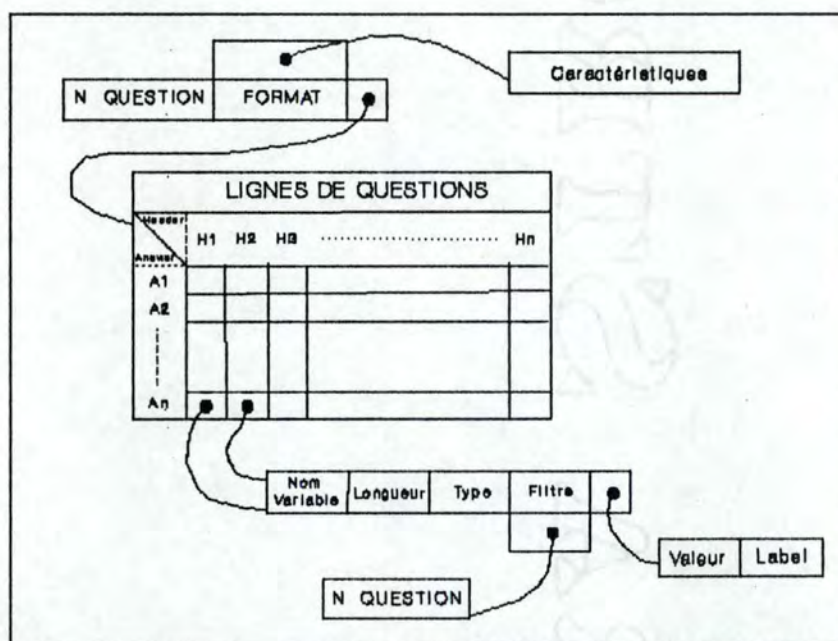


Fig. 4.1.1: Représentation d'une QUESTION

Etant donné la relation de dépendance qui existe entre les lignes de réponses et les lignes d'en-têtes, on a choisi de représenter cette relation par une matrice. A chaque élément de cette matrice sera associé une variable et chaque variable sera caractérisée par :

- * son nom,
- * sa longueur (nombre de 'digits' nécessaires),
- * son type (exemple : numérique, alpha-numérique),
- * sa nature (variable associée à une question filtre et le numéro de la question suivante non consécutive),
- * ses valeurs¹,
- * ses labels (label associé à chacune des valeurs)².

4.2 Le progiciel QUESTMAST

Nous nous limiterons par la suite à détailler le progiciel QUESTMAST³ avec lequel nous sommes plus familiers car il fut un de nos outils de travail lors de notre stage au CES. Ce programme a été utilisé pour produire les questionnaires relatifs à la "Scottish Young People's Survey".

La création de QUESTMAST a été rendue nécessaire à cause de la complexité de l'organisation et du stockage des informations d'une enquête postale. Non seulement la structure de ces informations, mais aussi la conception du questionnaire étaient rendues complexes par l'existence de différents types de questionnaires. Ainsi, les éléments qui ont conduit à concevoir ce logiciel générateur de questionnaires sont au nombre de trois : un désir de classifier et de gérer un grand nombre de questions, un désir d'uniformité sur la page et, enfin, un désir d'améliorer, voir d'automatiser, la production du système d'encodage. De plus, les programmeurs du CES ont décidé de greffer certaines fonctions à QUESTMAST telles que celle qui consiste à utiliser les informations générées dans les étapes de conception afin d'automatiser l'implémentation des différentes bases de données associées.

-
- 1 Les valeurs des variables sont destinés uniquement à l'analyse statistique.
 - 2 Les labels, tout comme les valeurs, sont utiles à l'analyse statistique.
 - 3 QUESTMAST est construit sur SIR (Scientific Information Retrieval) qui a été conçu par un centre du même nom, filiale de ISI. QUESTMAST suit rigoureusement la syntaxe définie dans SIR.

QUESTMAST répond donc à plusieurs fonctions dont deux sont essentielles. La première donne au concepteur de questionnaires les outils nécessaires à la construction d'une banque de questions, pour mettre en page ces questions, et pour produire les structures du questionnaire. La seconde fonction a pour but, à partir des informations générées dans la première fonction, de créer un schéma des différentes bases de données (base de données réelles et méta base de données).

Après une large description du fonctionnement de QUESTMAST ainsi que des différents mécanismes qui lui sont intégrés, nous tenterons, dans les chapitres suivants, d'expliquer la nécessité de documenter les bases de données.

4.2.1 La première fonction de QUESTMAST : la mise en page d'un questionnaire

Comme il a été dit précédemment, la question est un élément essentiel dans l'élaboration d'un questionnaire. Le contenu de la question ayant été défini par le chercheur, il reste au programmeur à produire la forme que cette question prendra sur la page du questionnaire. Pour effectuer cette tâche, un programmeur a, pour unique guide, l'apparence que se fait le chercheur de la question. L'objet de QUESTMAST, défini par le programmeur, consiste dès lors en la mise en forme des questions sur une page du questionnaire. Le mécanisme de mise en forme de QUESTMAST repose sur un jeu de règles et de contraintes.

Dans une première étape, nous allons examiner ces règles qui permettent donc un "formatage" des questions, tout en respectant les besoins et les requêtes des concepteurs.

Ces règles ont été regroupées en trois classes : les éléments appartenant à la première classe déterminent la représentation même de la question alors que les éléments des deux autres classes concernent la production d'informations-clés nécessaires à l'élaboration des différentes bases de données. Pour illustrer ce propos, considérons le fichier de questions **MASTERFILE**.

Le fichier de questions MASTERFILE (Fichier Maître) est construit en utilisant un éditeur de texte externe à QUESTMAST. On y inscrit les questions en respectant premièrement les règles de mise en page (celles-ci correspondent aux règles de la première classe) :

- * ~5texte~ signifie que "texte" sera en gras
- * [texte] signifie que "texte" sera souligné
- * <texte> signifie que "texte" sera en italique
- * texte \$texte signifie que "texte" qui suit le symbole \$ se trouvera à la ligne suivante
- * texte \$_\$texte signifie que l'on insérera une ligne de blancs entre le premier "texte" et le "texte" qui suit la séquence de symboles \$_\$

Il faut encore tenir compte :

- d'une part de la série de règles de la deuxième classe selon lesquelles le caractère de la première colonne doit être obligatoirement :

Q Pour représenter le texte de la Question.
Le texte relatif à une question peut dépasser une ligne, mais chaque nouvelle ligne doit obligatoirement avoir un blanc en première colonne.

A Pour représenter une réponse à la question (Answer).
Si il y a plusieurs réponses possibles, cela implique une série de A en première colonne.

H Pour représenter un en-tête de la question (Header).
Un en-tête est de la forme suivante :

	oui	non	indécis
... vous a aidé à obtenir un grade	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Si il y a plusieurs en-têtes possibles, cela implique une série de H en première colonne.

- d'autre part de la série de règles de la troisième classe selon lesquelles le caractère de la première colonne doit être obligatoirement :

- D Pour représenter une ligne qui reprend, entre autres, le numéro de la question et son format.
- V Pour représenter une ligne qui reprend le nombre de variables associées à la question ainsi que les noms et bornes de ces variables.

Dans une seconde étape, ces règles vont être interprétées par QUESTMAST pour produire la mise en forme des questions.

En voici un exemple :

Soit le fichier MASTERFILE suivant :

```

D1. 22
Q"5Avez-vous quitté l'école ?"
HOU
HNON
A<Si non, allez à la question n 3>
VCOLS=1 NAME=QUITECOL RANGE=1-4

D2. 2
Q"5Pourquoi avez-vous quitté l'école ?"
$<cocher toutes les cases convenant>
A..... parce que j'avais un boulot
A..... parce que j'en avais marre de l'école
A..... parce que j'avais assez de connaissances
VCOLS=3 NAME=PQUITTE1-PQUITTE3 RANGE=1-2

D3. 48
Q"5Regrettez-vous votre décision ?"
$<cocher UNE case>
Aoui.....
Anon.....
A pas sûr.....
VCOLS=1 NAME=REGRET RANGE=1-3

```

QUESTMAST traitera ce fichier pour en produire le questionnaire suivant :

1. Avez-vous quitté l'école ?

OUI

☐ 1

NON

☐ 2


Si non, allez à
la question n 3

Si oui :

2. Pourquoi avez-vous quitté l'école ? (cocher toutes les cases convenant)

..... parce que j'avais un boulot

☐

..... parce que j'en avais marre de l'école

☐

..... parce que j'avais assez de connaissances

☐

3. Regrettez-vous votre décision ? (cocher UNE case)

oui

☐ 1

non

☐ 2

pas sûr

☐ 3

La DLIGNE est relativement importante, parce qu'elle reprend le **NUMERO DE FORMAT** que l'on attache à la question. Ce système d'attribution de numéro de format à chaque question répond à l'objectif d'uniformité sur une page.

Le texte d'une question est stocké dans une structure non formatée et le numéro de format détermine l'apparence finale de la question sur la page et également le nombre de réponses possibles.

Soient les deux questions :

D1. 2

Q Que faites-vous actuellement ? (cochez UNE SEULE case)

A Je travaille à temps plein

A J'étudie à temps plein

A Je suis des cours de formation

A Autre

D2. 8

Q Parmi ces activités, lesquelles avez-vous exercées ?

(cochez TOUTES les cases qui conviennent)

A Un travail à temps plein

A Des études à temps plein

A Des cours de formation

Bien que le texte de ces deux questions soit relativement semblable, les réponses attendues sont différentes : on attend une seule réponse à la question D1 alors que l'on en attend au maximum trois dans la question D2. L'attribution de différents numéros de format indique à QUESTMAST quel type de réponses implique la question posée. On adoptera la convention suivante : les cases de la question D1 seront numérotées alors que celles de la question D2 ne le seront pas.

4.2.2 La seconde fonction de QUESTMAST : la création d'un schéma de base de données

Il apparaît que le format représente une notion importante. Par l'attribution d'un numéro de format, on classe les questions prioritairement suivant leur caractéristique statistique et secondairement selon leur caractéristique de saisie. Il est possible ensuite d'identifier ces caractéristiques à un plus haut niveau et ainsi de déduire les attributs de chaque variable.

L'implémentation de QUESTMAST suit ce principe, mais s'intéresse à une analyse des caractéristiques des formats. Nous pouvons illustrer la démarche suivie par QUESTMAST par un exemple.

Reprenons le questionnaire :

1. Avez-vous quitté l'école ?

OUI 1 NON 2 → Si non, allez à la question n 3

Si oui :

2. Pourquoi avez-vous quitté l'école ? (cocher toutes les cases convenant)

..... parce que j'avais un boulot ☐
 parce que j'en avais marre de l'école ☐
 parce que j'avais assez de connaissances ☐

3. Regrettez-vous votre décision ? (cocher UNE case)

oui 1
 non 2
 pas sûr 3

A partir de ce questionnaire, quelles informations sont extraites de manière automatique par QUESTMAST ?

QUESTMAST produit le schéma suivant :

Record Schema 1

Data List	Fixed (1)		
/1	Caseld	8-11	(1)
/1	Qultecol	12-13	(1)
/1	Pqultte1	14-15	(1)
/1	Pqultte2	16-17	(1)
/1	Pqultte3	18-19	(1)
/1	Regret	20-21	(1)

Var Ranges

Qultecol	(1 4) /
Pqultte1	(1 2) /
Pqultte2	(1 2) /
Pqultte3	(1 2) /
Regret	(1 3) /

End Schema 1

4.2.3 Les autres fonctions intégrées dans QUESTMAST

Une fois que les questions ont été introduites dans la banque de questions, l'utilisateur peut définir des pages thématiques qui sont alors stockées dans une MATRICE DE COMPOSITION. La définition d'une page consiste en une série ordonnée de numéros de questions. Un nom est donné à chaque page qui peut aussi posséder plusieurs versions. Plusieurs pages peuvent appartenir à un même thème, et auront donc un même nom, mais elles peuvent contenir des questions différentes, ce qui permet à l'utilisateur de construire les versions en fonction des sujets qu'il veut traiter.

Dans les paragraphes qui suivent, nous nous proposons de donner une description plus fine de la MATRICE DE COMPOSITION et de son utilisation.

Supposons que nous avons un échantillon d'étudiants qui sont dans leur année terminale obligatoire (élèves âgés de plus ou moins 16 ans). Ces élèves peuvent décider soit de quitter l'école, soit de continuer leurs études. On divise alors cette population en deux sous-groupes, appelés respectivement les "Quittants" (Leavers) et les "Restants" (Stayers), auxquels on envoie des questionnaires différents.

Rappelez-vous le schéma représentant les échantillons des enquêtes⁴ qui suggère deux types de quittants, les individus qui quittent soit après la S4, soit après la S5 et deux types de restants, les élèves qui continuent après la S4 et après la S5.

Il est évident que certaines questions posées aux individus qui continuent en S5 et S6 seront différentes de celles posées à ceux qui quittent après S4. De même, les questionnaires adressés à ceux qui quittent sont souvent orientés vers l'activité (le travail) alors que pour ceux qui ne cessent pas leurs études, ils sont plutôt orientés vers les possibilités d'études supérieures.

C'est pour ces raisons que l'on élabore quatre types de questionnaires : L1, L2, S1, S2.

Voici une description de la MATRICE DE COMPOSITION :

PAGE NAME	VERSIONS	
Lieu d'activité	I	II
Quittant l'école	I	II
Cours actuels	I	
Restant	I	II
Situation sociale	I	

Ceci nous apprend qu'il y a donc cinq pages dans la matrice de composition. Trois de ces cinq pages existent en deux versions. On peut alors construire les quatre questionnaires L1, L2, S1, S2 :

⁴ Cfr Chapitre 2, section 2.1.3

- L1 : Page 1 : Lieu d'activité (I)
 Page 2 : Quittant l'école (II)
 Page 3 : Situation sociale (I)
 L2 : Page 1 : Lieu d'activité (II)
 Page 2 : Quittant l'école (I)
 Page 3 : Situation sociale (I)
 S1 : Page 1 : Lieu d'activité (I)
 Page 2 : Cours actuels (I)
 Page 3 : Restant (I)
 Page 4 : Situation sociale (I)
 S2 : Page 1 : Lieu d'activité (II)
 Page 2 : Cours actuels (I)
 Page 3 : Restant (II)
 Page 4 : Situation sociale (I)

En outre, on utilise encore le numéro de format pour déterminer la nature des variables associées à la question et ainsi l'information appropriée que l'on insérera dans les bases de données. Pour chaque numéro de format, on peut construire une TABLE DE DEDUCTION qui sera utilisée pour générer les informations des différentes bases de données. Par exemple, supposons que le format n°2 soit attribué à la question D1 et le format n°8 à la question D2, on obtient la table :

FORMAT	NBVARs	BORNE_INF	BORNE_SUP
2	1	1	a
8	a	1	2

Ce qui signifie que le format n°2 requiert une seule variable et les bornes inférieure et supérieure ont respectivement les valeurs 1 et 'a' (où 'a' est le nombre de ALIGNE). Le format n°8 requiert, quant à lui, 'a' variables et chacune de ces variables a des bornes de 1-2 (c'est-à-dire la case est cochée ou elle ne l'est pas). Dans le cas de la question D1, 'a' prendrait la valeur 4 et pour la question D2, 'a' vaudrait 3.

5. DESCRIPTION DES MODELES D'ANALYSE FONCTIONNELLE DU SYSTEME D'INFORMATION

Dans ce cinquième chapitre, nous tenterons de modéliser les informations ainsi que leur flux pour le cas spécifique d'une enquête par questionnaire, en partant du cas le plus général (section 5.1) pour aboutir au cas particulier du CES (section 5.3).

La méthodologie d'analyse utilisée pour décrire ce cas est la méthodologie d'analyse fonctionnelle enseignée à l'Institut d'Informatique des Facultés Notre Dame de la Paix de Namur.

Dans la section 5.3 sera exposé le modèle Entité/Association (E/A)¹ relatif au système d'enquête du CES. Nous procéderons ensuite dans la section 5.4 à la transformation de ce modèle E/A en un Modèle d'Accès Généralisé (MAG)². Afin d'en faciliter la lecture, nous nous proposons au préalable, dans la section 5.2, d'exposer certains concepts propres à ce type d'analyse, à savoir la dynamique des traitements et le diagramme des flux.

5.1 Le modèle général des opérations de traitement des informations d'une enquête par questionnaire

Il a été observé, non seulement au CES mais aussi dans d'autres centres, que la variété des opérations composant une enquête par questionnaire, est importante. La multitude des événements nous amène tout naturellement à opérer un regroupement par domaine commun afin de mettre en évidence les agents-clés intervenant dans l'élaboration d'une telle enquête.

Ces agents sont au nombre de trois : la recherche, l'informatique et la récolte des informations. Ces différents agents sont en relation les uns avec les autres. Ces interdépendances sont illustrées par le schéma de la page suivante :

-
- 1 Le modèle E/A a été développé par Y. PIGNEUR et F. BOBART, chargé de cours à l'Institut d'Informatique.
 - 2 Le modèle MAG a été décrit par J.-L. HAINAUT, chargé de cours à l'Institut d'Informatique.

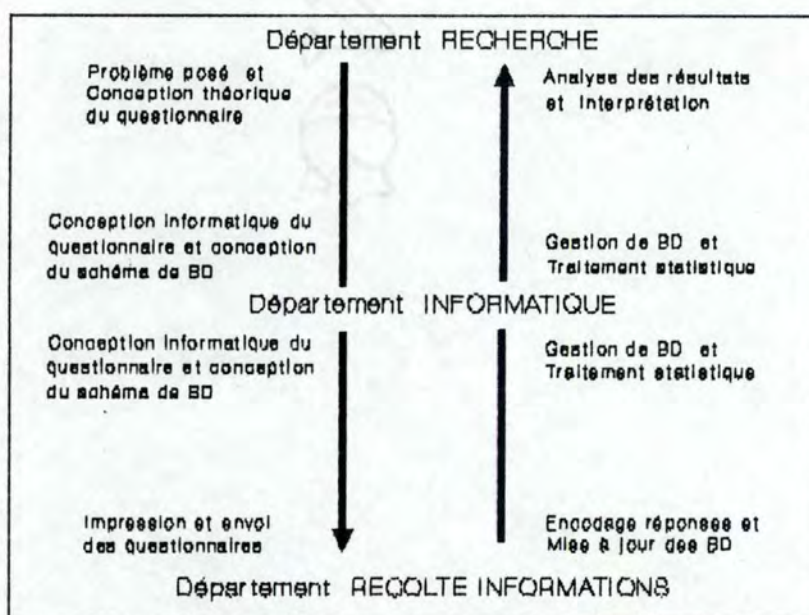


Fig. 5.1.1 : Schéma général du système d'information

Le département RECHERCHE analyse, dans un premier temps, un problème qui lui est posé, afin de pouvoir déterminer si il sera nécessaire de mettre en oeuvre une ou plusieurs études pour répondre à cette question.

Chacune de ces études fera l'objet d'une nouvelle analyse dans le but de définir le nombre et le type d'enquête qui lui sera associé. En s'appuyant sur la théorie des enquêtes par questionnaire³, les chercheurs reprennent chaque enquête en particulier afin d'en déterminer les deux éléments prépondérants, à savoir l'échantillon et le questionnaire.

En ce qui concerne la détermination de l'échantillon, les chercheurs sont seuls concernés. Quant à l'élaboration du questionnaire, le département RECHERCHE fait aussi appel au département INFORMATIQUE.

³ La théorie des enquêtes par questionnaire est décrite au chapitre 3

En effet, les chercheurs conçoivent le questionnaire de manière théorique et la tâche des programmeurs du département INFORMATIQUE consiste en la mise en page du questionnaire. Lorsque l'on dispose de la forme finale et définitive du questionnaire, les programmeurs créent les différents schémas de base de données.

Par la suite, le questionnaire définitif est transmis au département RECOLTE INFORMATIONS qui réalise l'impression du questionnaire et qui est responsable de l'envoi de celui-ci aux différents individus qui composent l'échantillon.

Quand les questionnaires remplis ont été retournés au CES, le travail des encodeurs peut commencer. Ils encodent les réponses et mettent à jour les différentes bases de données. A cette fin, nous avons conçu un programme baptisé "Coding Notes". En effet, il nous a été donné de constater que certaines phases d'encodage étaient encore effectuées manuellement. Dès lors, notre objectif a été, d'une part, d'élaborer un schéma de base de données, susceptible de recueillir les informations relatives à l'encodage des questionnaires et, d'autre part, de créer des programmes d'extraction de ces informations, dont le principe de base est l'analyse et la reconnaissance de texte.

La phase d'encodage terminée, les programmeurs traitent les données encodées pour qu'elles adoptent une forme plus "statistique" car ces données doivent pouvoir être manipulées par un logiciel statistique, tel que par exemple SPSS. Ils remanient aussi la structure des bases de données afin que celles-ci soient tout à fait au point pour une consultation aisée des données.

La dernière étape relative à l'analyse et à l'interprétation des résultats est réservée aux chercheurs.

Cette analyse permet à ceux-ci de réfuter ou d'accepter les hypothèses de travail posées en début d'étude.

5.2 Les modèles de traitement des informations

5.2.1 La dynamique des traitements

Suivant la théorie de la dynamique des traitements conçue par Messieurs F. BODART et Y. PIGNEUR et grâce au schéma général des informations d'une enquête par questionnaire décrit dans la section précédente, on peut à présent déterminer de manière plus précise les événements et les conditions nécessaires à la réalisation d'une enquête.

On peut affirmer que l'évènement source est un problème observé que l'on désire approfondir, voire même expliquer, grâce à une enquête. Il se présente alors trois cas : soit le problème est dit faisable ou recevable, soit il a déjà été réalisé ou, au pire, il est irréalisable. Dans le cas de faisabilité, on entre directement dans le processus d'ENQUETE, dans le cas contraire le processus n'est pas déclenché.

L'enquête proprement dite est le premier événement d'une cascade de trois opérations, les deux autres étant la détermination de l'ECHANTILLON et l'élaboration du QUESTIONNAIRE sous sa forme définitive⁴ (point 1 de la figure 5.2.1).

Le questionnaire réalisé, deux processus peuvent être déclenchés en parallèle : l'impression du questionnaire et la conception des différentes bases de données et écrans de saisie (points 2 et 3 de la figure 5.2.1). Quand le premier de ces deux processus est terminé, il est possible d'envoyer les questionnaires aux individus de l'échantillon.

Il faudra attendre le retour de tous les questionnaires avant de poursuivre notre dynamique des traitements. La fin de la conception des bases de données et la réception de tous les questionnaires forme un point de synchronisation qui conduit à la phase d'encodage (point 4 de la figure 5.2.1). Il ne sera possible de traiter les résultats (point 5 de la figure 5.2.1) que si toutes les réponses sont encodées.

⁴ Ces trois éléments ont été décrits au chapitre 3, section 3.1.

Les évènements décrits précédemment donnent lieu au schéma suivant :

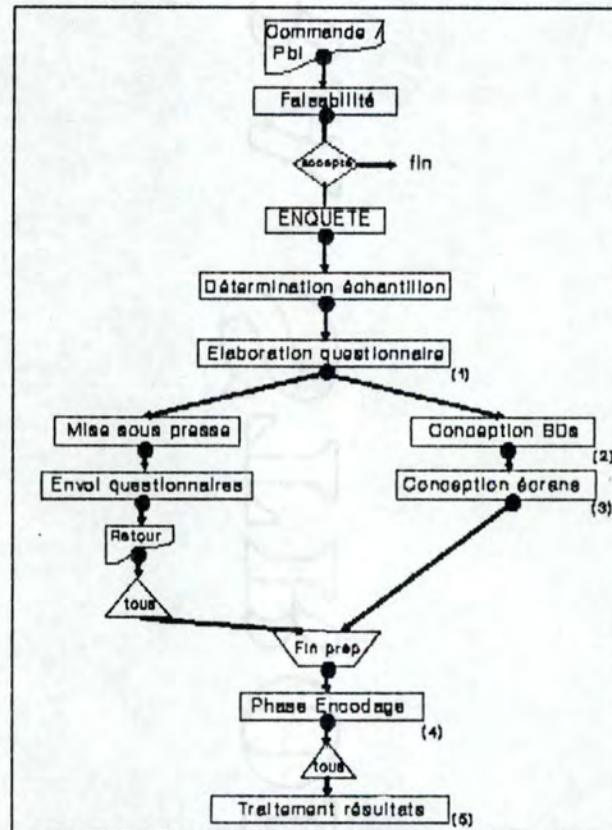


Fig. 5.2.1: Schéma de la dynamique des traitements

Afin de poursuivre la description du processus d'enquête, nous proposons dans la section suivante de nous intéresser aux opérations que l'on pourrait qualifier de plus "informatiques" en considérant le diagramme des flux. Les opérations informatiques sont avant tout centralisées dans le département INFORMATIQUE et dans le département RECOLTE INFORMATIONS.

5.2.2 Le diagramme des flux

Le schéma du diagramme des flux présenté ci-après couvre donc uniquement les évènements (1), (2), (3), (4) et (5) du schéma de la dynamique des traitements :

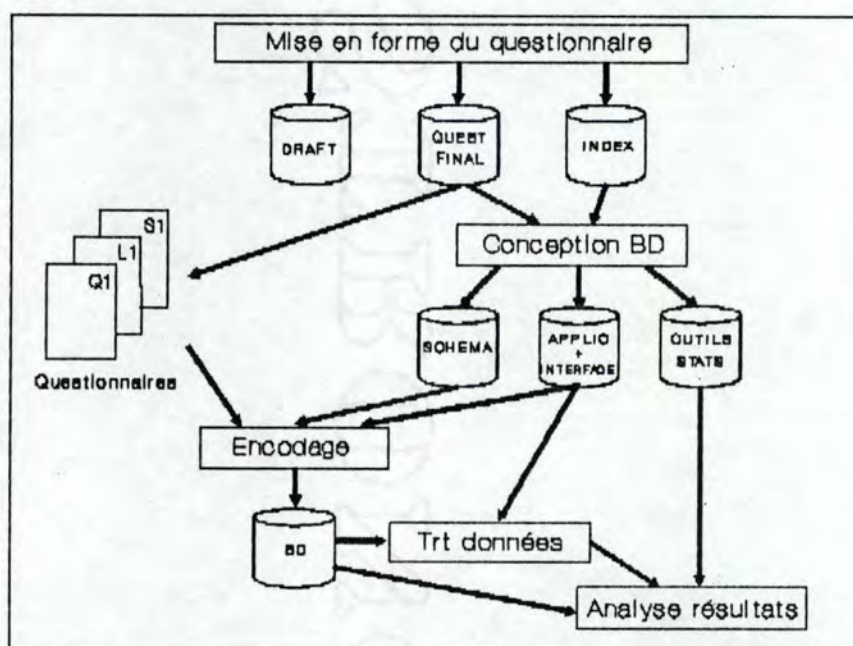


Fig. 5.2.2 : Schéma du diagramme des flux

Nous avons signalé que les opérations de mise en forme du questionnaire, de conception des bases de données et d'encodage sont trois des opérations "informatiques".

La première de ces opérations se réalise, pour le CES, grâce à QUESTMAST. Ce logiciel produit trois types d'outputs : le draft (connu encore sous le nom de MASTERFILE), le questionnaire définitif et un fichier index qui contient des informations relatives aux questions (exemple : numéro de la question, format, nom de la variable correspondante). Le questionnaire final sera utilisé pour produire les questionnaires à envoyer. Associé au fichier index, le questionnaire définitif détermine la deuxième opération : la conception des bases de données.

Cette opération entraîne la création d'autres éléments tels qu'un schéma, des applications-interface et des outils statistiques.

Les questionnaires retournés et la deuxième opération terminée induisent la dernière opération qui est l'encodage dans des bases de données qui sont nécessaires au traitement des données et à l'analyse des résultats.

5.3 Le modèle Entité/Association des informations

Après analyse des modèles de traitements, nous avons donc pu construire un schéma Entité/Association¹, qui indique sous forme visuelle le contenu sémantique du problème ainsi que les inter-relations qui peuvent exister entre les différents constituants ou concepts.

Pour rappel et sans entrer dans le détail, ce schéma se base sur les principes suivants.

Tout objet ou concept individualisé du réel est dénommé Entité. Les entités sont répertoriées en classes appelées Types d'Entité (T.E). Elles peuvent être en association les unes avec les autres. Ces associations sont, quant à elles, groupées en Types d'Association.

Quant aux propriétés des objets du réel, elles sont représentées par des Attributs. Ceux-ci sont reliés aux types d'entité et aux types d'association qu'ils symbolisent.

Graphiquement, ces différentes composantes se représentent de façon distincte. Une entité correspond à une boîte étiquetée au nom de l'entité. Une association correspond à un losange relié à deux entités. Les attributs sont disposés sous les types d'entité ou d'association auxquels ils se rapportent.

Enfin, par un couple de chiffres apposé sur les lignes reliant des entités, on indique le nombre de fois qu'une entité participe à l'association qui lui correspond. On parle alors de Connectivité, c'est-à-dire le nombre minimum et maximum de participations d'une entité à une association.

5.3.1 Les entités

Après observation des éléments composant le flux d'informations, nous avons pu distinguer les types d'entités suivants :

1 Ce schéma est décrit par F. BODART et Y. PIGNEUR.

T.E ETUDE : Une ETUDE est une entité qui reprend aussi bien des études se produisant une seule fois que des études longitudinales ("cohort"). Une ETUDE peut également être déterminée à partir d'études existantes.

Une étude n'adopte que deux statuts : soit le statut de définition, soit le statut de réalisation. Le premier statut représente le stade THEORIQUE de l'étude alors que le second statut est adopté lorsque les analyses de résultats ont été effectués c'est-à-dire un stade plus "PRATIQUE".

ATTRIBUTS : IDETUD : Identifiant de l'étude ,
 NOMETUD : Nom complet et significatif de l'étude,
 ANNEETUD : Année de l'étude,
 TYPETUD : Type de l'étude
 (ex : "Cohort", "cross-sectional"),
 NATURETUD : Nature de l'étude
 (ex : stade de définition, stade de réalisation),
 NUMPHASE : Numéro de la phase ("sweep") de l'étude longitudinale.

IDENTIFIANT : IDETUD.

T.E ENQUETE : Une ENQUETE est une entité dont l'objectif est la mise en évidence de données statistiques. Pour réaliser ce but, une ENQUETE utilisera comme outils, des QUESTIONNAIRES et portera sur un ECHANTILLON représentatif d'un ensemble de jeunes. Une ENQUETE est conçue par un chercheur suite à la demande d'un commanditaire (organisme public ou privé).

ATTRIBUTS : IDENQ : Identifiant de l'enquête,
 NOMENQ : Nom complet et significatif de l'enquête (8 à 10 caractères),
 TITRENQ : Titre de l'enquête,
 ANNEENQ : Année de l'enquête,
 PERIODENQ : Période de l'enquête
 (ex : automne, printemps),
 DATDEBUT : Date d'émission de l'enquête,
 DATFIN : Date de clôture de l'enquête,

COMMAND : Commanditaire de l'enquête.

IDENTIFIANT : IDENQ.

T.E POPULATION : Une POPULATION est une entité qui contient tous les individus qui ont fait ou qui feront l'objet d'un échantillonnage.

ATTRIBUTS : IDPOP : Identifiant de la population,
COHOPOP : Caractéristiques de la population,
LOCALPOP : Localisation de la population
(ex : Grampian, Fife (deux régions d'Ecosse)).

IDENTIFIANT : IDPOP.

T.E ECHANTILLON : Un ECHANTILLON est une entité qui regroupe des individus qui ont subi une technique d'échantillonnage. Il faut à tout moment garder à l'esprit que cet échantillon doit être représentatif de toute une population.

ATTRIBUTS : IDECHANT : Identifiant de l'échantillon,
TYPECHANT : Type de l'échantillon,
DATECHANT : Date de la construction de l'échantillon,
METHODE : Méthode d'échantillonnage,
CRITERE : Critère d'échantillonnage.

IDENTIFIANT : IDECHANT.

T.E SOUS-ECHANTILLON : Un SOUS-ECHANTILLON est une entité similaire à l'entité ECHANTILLON, mais de taille moindre.

ATTRIBUTS : IDSSECH : Identifiant du sous-échantillon,
TYPSSSECH : Type du sous-échantillon,
CRITERE : Critère du sous-échantillonnage.

IDENTIFIANT : IDSSECH.

T.E QUESTIONNAIRE : Un QUESTIONNAIRE est une entité qui regroupe toutes les QUESTIONS que le chercheur désire insérer dans l'ENQUETE en vue d'une analyse statistique répondant au problème qu'il s'est posé dans son ETUDE. Il est possible de découper un QUESTIONNAIRE en PAGES.

ATTRIBUTS : IDQUAIRE : Identifiant du questionnaire,
DATCREAT : Date de création.

IDENTIFIANT : IDQUAIRE.

T.E QUESTION : Une QUESTION se compose d'un texte (cfr Q-Ligne), de réponses possibles (cfr A-Ligne) et d'en-têtes (cfr H-Ligne).

ATTRIBUTS : IDQUEST : Identifiant de la question.

IDENTIFIANT : IDQUEST.

T.E VARIABLE : Une VARIABLE est une entité qui répertorie toutes les informations disponibles au sujet d'une VARIABLE à analyser. Une VARIABLE est associée à une QUESTION. La valeur de cette VARIABLE sera la donnée statistique proprement dite et possède des limites inférieure et supérieure. Il est à signaler qu'une VARIABLE peut dépendre d'une autre variable (par exemple, dans le cas d'une question-filtre).

ATTRIBUTS : IDVAR : Identifiant de la variable,
NOMVAR : Nom significatif de la variable,
LABVAR : Label de cette variable,
NMIN : Valeur minimale de la variable,
NMAX : Valeur maximale de la variable,
TYPEVAR : Type de la variable
(ex : numérique, alphabétique),
GROUPVAR : Groupe auquel appartient cette variable
(ex : PQUITTE1-PQUITTE3)².

² Cfr Chapitre 4, section 4.2.1.

VARTXT : Texte provenant de la question du questionnaire et qui est associé à la variable.

VALTXT : Texte provenant des réponses et en-têtes du questionnaire et qui représente la valeur de la variable.

IDENTIFIANT : IDVAR et l'association POSSEDE avec QUESTION.

T.E BASEDON : Une BASE de DONNEES est une entité qui contient principalement les valeurs des variables, mais encore d'autres informations (ex: ADMIN³).

ATTRIBUTS : IDBD : Identifiant de la base de données,
 NOMFICH : Nom complet du fichier de la base de données,
 TYPEFICH : Type du fichier de la base de données
 (ex : fichier SIR, fichier SPSS),
 STATUTBD : Statuts de la base de données
 (ex : archive, on-line, bande,...),
 DATMAJ : Date de la dernière mise à jour de la base de données,
 TAILLE : Taille de la base de données,
 NBVARS : Nombre de variables contenues dans la base de données,
 NBCASES : Nombre de 'cases' contenues dans la base de données,
 ACCES : Niveau de sécurité d'accès à la base de données,
 NUMCAT : Numéro du catalogue où se trouve la base de données.

IDENTIFIANT : IDBD.

T.E RESPONSABLE : Un RESPONSABLE est une entité qui contiendra toutes les personnes qui interviennent, à quelque niveau que ce soit, dans le processus d'élaboration d'une

ENQUETE. Il peut être un programmeur, un concepteur de base de données, un chercheur ou un créateur.

ATTRIBUTS : IDRESP : Identifiant du responsable,
NOMRESP : Nom du responsable,
PRENOMRESP : Prénom du responsable,
TITRESP : Titre du responsable,
ORGRESP : Organisme, institution ou
centre auquel appartient
ce responsable,
ADRESRESP : Adresse de cet organisme.

IDENTIFIANT : IDRESP.

T.E PAGE/THEME : Une PAGE rassemble les QUESTIONS relatives à un même thème. Par exemple, toutes les questions relatives à l'enseignement primaire et secondaire seront reprises sous le label de 'Formation de Base'.

ATTRIBUTS : IDPAGE : Identifiant de la page,
NOMPAGE : Nom significatif de la page,
THEME : Thème de la page (titre).

IDENTIFIANT : IDPAGE.

T.E TYPQUEST : Un TYPE de QUESTION est une entité qui permettra de distinguer les différents types de questions utilisées. Si on se réfère à la description des étapes générales d'une enquête, il existe au moins quatre types de questions : ouverte, fermée, semi-fermée, filtre.

ATTRIBUTS : IDTYPE : Identifiant du type de question,
NOMTYPE : Nom du type (filtre, fermée, ...),
NUMTYPE : Numéro du type de question.

IDENTIFIANT : IDTYPE.

T.E FORMAT : Un FORMAT est une entité qui déterminera la forme que prendra la QUESTION sur la PAGE du QUESTIONNAIRE. Un FORMAT sera donc une fonction dont les arguments principaux seront : le type de question, le nombre de réponses possibles et les en-têtes de choix.

ATTRIBUTS : IDFORMAT : Identifiant du format,
NUMFORMAT : Numéro du format,
TXTFORMAT : Description du format utilisé,
VALMAXVAR : Valeur maximale de la variable,
VALMINVAR : Valeur minimale de la variable,
NBQLIGNE : Nombre de QLIGNE de la variable,
NBALIGNE : Nombre de ALIGNE de la variable,
NBHLIGNE : Nombre de HLIIGNE de la variable.

IDENTIFIANT : IDFORMAT.

T.E QLIGNE : Une QLIGNE est une entité qui contient les lignes de texte de la QUESTION.

ATTRIBUTS : IDQLIGNE : Identifiant de la QLIGNE,
NUMQLIGN : Numéro de la QLIGNE,
TXTQLIGN : Texte de la QLIGNE.

IDENTIFIANT : IDQLIGNE.

T.E ALIGNE : Une ALIGNE est une entité qui contient toutes les lignes de réponses possibles, c'est-à-dire toutes les réponses proposées dans les questions fermées ou semi-fermées du QUESTIONNAIRE.

ATTRIBUTS : IDALIGNE : Identifiant de la ALIGNE,
NUMALIGN : Numéro de la ALIGNE,
TXTALIGN : Texte de la ALIGNE.

IDENTIFIANT : IDALIGNE.

T.E HLINE : Une HLINE est une entité qui contient tous les en-têtes utilisés pour les différentes cases de choix.

ATTRIBUTS : IDHLINE : Identifiant de la HLINE,
 NUMHLIGN : Numéro de la HLINE,
 TXTHLIGN : Texte de la HLINE.

IDENTIFIANT : IDHLINE.

T.E VALMQT : Une VALMQT est une entité qui regroupe toutes les valeurs manquantes correspondant à certaines QUESTIONS. On attribuera, par exemple, la valeur 99 à une variable si l'individu n'a pas répondu à la question.

ATTRIBUTS : IDVMQT : Identifiant de la valeur
 manquante,
 VALVMQT : Valeur numérique de la valeur
 manquante,
 LABVMQT : Label de cette valeur numérique.

IDENTIFIANT : IDVMQT.

T.E FREQ : Une FREQ est une entité spéciale qui reprend toutes les fréquences associées à une variable. Ces fréquences sont soit relatives, soit absolues.

ATTRIBUTS : IDFREQ : Identifiant de la fréquence,
 LABFREQ : Label relatif à la fréquence,
 FREQREL : Fréquence relative,
 FREQABS : Fréquence absolue.

IDENTIFIANT : IDFREQ.

T.E CATEGORIE : Une CATEGORIE est une entité qui contient toutes les caractéristiques (facteurs) qui sont les bases de la post-stratification.

ATTRIBUTS : IDCAT : Identifiant de la catégorie,

FACTCAT : Facteur de catégorie
(ex: âge, sexe, résultat scolaire),
NBCAT : Nombre de catégories.

IDENTIFIANT : IDCAT.

T.E LOCALISATION : Une LOCALISATION est une entité qui permettra de mentionner à l'utilisateur de la base de données, le type de MACHINE associé à la base de données. MACHINE signifie ici le hardware (ex: VAX, SUN, PC, ...).

ATTRIBUTS : IDMACH : Identifiant de la machine,
TYPEMACH : Type de la machine,
OPSYS : Système d'exploitation (OS) installé
sur la machine,
OUMACH : Localisation de la machine,
TXTMACH : Notes explicatives relatives à la
machine.

IDENTIFIANT : IDMACH.

T.E NOTE_ENQ : Une NOTE_ENQ est une entité qui offre une note explicative sur l'entité ENQUETE.

ATTRIBUTS : IDNOTENQ : Identifiant de la note,
TXTENQ : Texte de la note.

IDENTIFIANT : IDNOTENQ.

T.E NOTE_VAR : Une NOTE_VAR est une entité qui offre une note explicative sur l'entité VARIABLE.

ATTRIBUTS : IDNOTVAR : Identifiant de la note,
TXTVAR : Texte de la note.

IDENTIFIANT : IDNOTVAR.

T.E NOTE_BD : Une NOTE_BD est une entité qui offre une note explicative sur l'entité BASEDON.

ATTRIBUTS : IDNOTBD : Identifiant de la note,
TXTBD : Texte de la note.

IDENTIFIANT : IDNOTBD.

5.3.2 Les relations

Les associations

Dans un premier temps, il semble utile de préciser les caractéristiques de certaines associations entre entités. En effet, ces associations ont la particularité d'avoir des attributs.

L'association COMPORTE (entre les entités QUESTION et PAGE/THEME) possède deux attributs :

- * VERSION ('Wording') : donne la version de la question telle qu'elle est reprise sur la page,
- * POSITION : donne la position de la question sur la page.

Identifiant : VERSION et la relation avec l'entité PAGE.

L'association DEPEND (entre l'entité VARIABLE et VARIABLE) possède un attribut :

- * SOURCE : donne la source de la variable, c'est-à-dire son origine (ex: provient d'un questionnaire, dérivée d'une autre variable).

Identifiant : Les relations avec l'entité VARIABLE.

L'association EXTRAIT (entre les entités POPULATION et ECHANTILLON) possède un attribut :

- * PROPORTION : donne la proportion de l'échantillon dans la population.

Identifiant : La relation avec l'entité POPULATION, la relation avec l'entité ECHANTILLON et PROPORTION.

L'association LIAISON (entre l'entité ETUDE et ETUDE) possède un attribut :

- * NATURE : donne la nature de la liaison entre les études. Soit l'étude est la réalisation de la définition de l'étude, soit l'étude se base sur une définition ou une réalisation d'étude.

Identifiant : Les relations avec l'entité ETUDE.

L'association FILTRE (entre l'entité QUESTION et l'entité VARIABLE) possède un attribut :

- * VALEUR : donne la valeur qui représente la question vers laquelle on est renvoyé.

Identifiant : La relation avec l'entité VARIABLE et VALEUR.

Les contraintes supplémentaires

Dans un second temps, nous spécifions certaines contraintes supplémentaires; celles-ci consistent en des conditions que les attributs des différentes entités doivent respecter afin d'obtenir une cohérence parfaite des schémas Entité/Association :

- * QUESTIONNAIRE (:DATCREAT) >= ECHANTILLON (:DATECHANT),
- * QUESTIONNAIRE (:DATCREAT) >= ENQUETE (:DATDEBUT),
- * ECHANTILLON (:DATECHANT) >= ENQUETE (:DATDEBUT),
- * ENQUETE (:DATFIN) >= ENQUETE (:DATDEBUT).

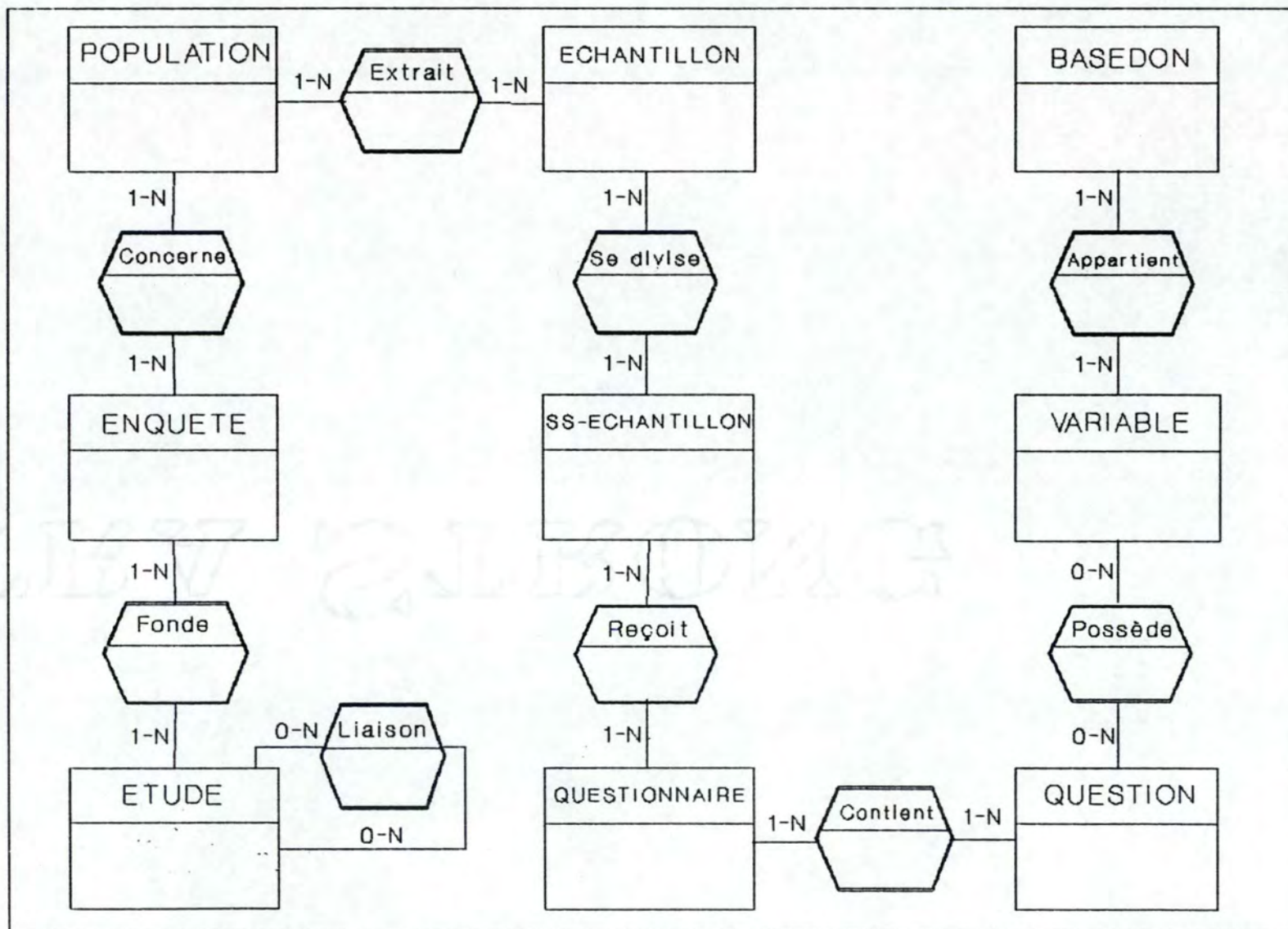


Fig. 5.3.1 : Schéma Entité / Association général

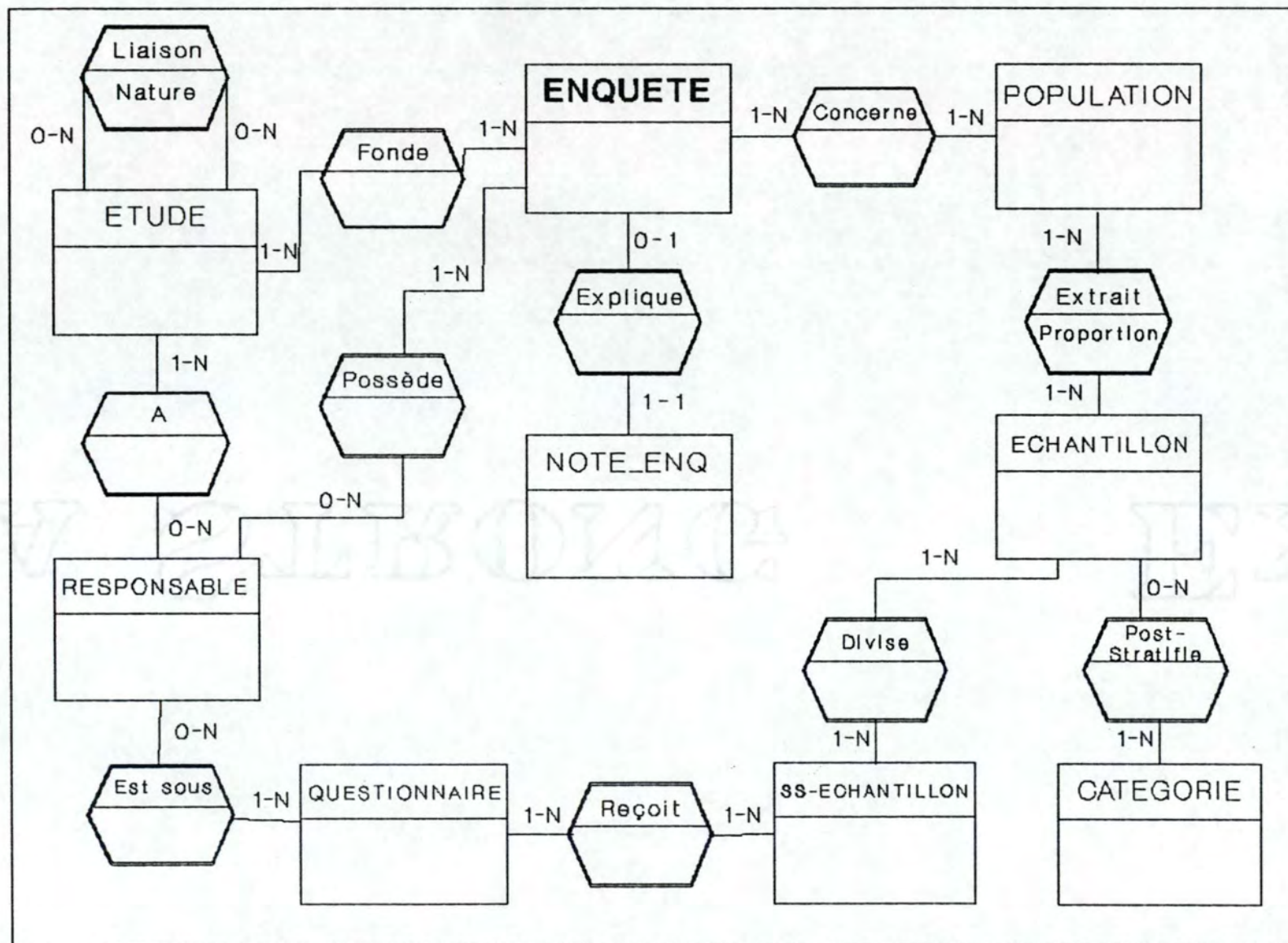


Fig. 5.3.2 : Sous-Schéma pour l'ENQUETE

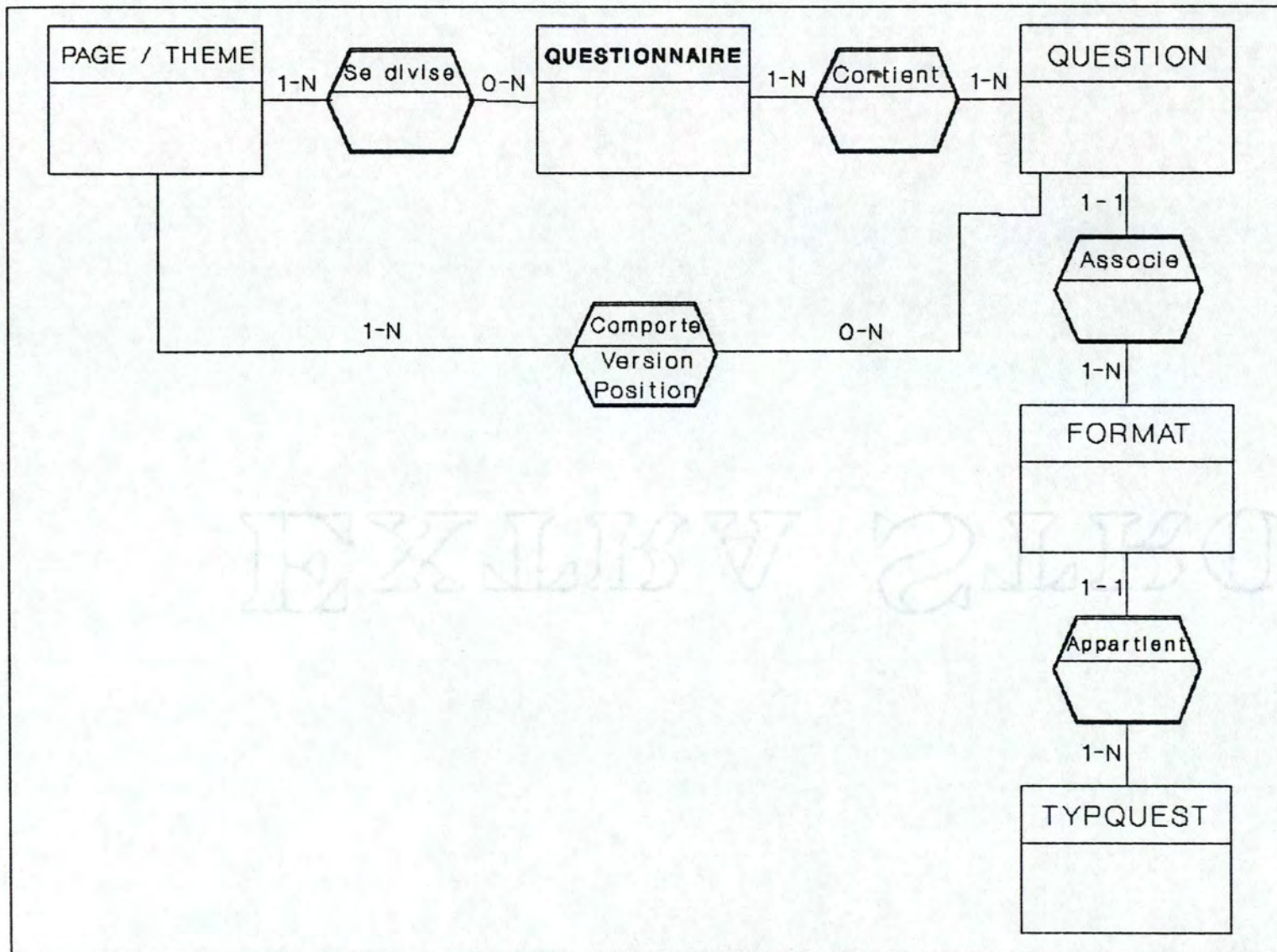


Fig. 5.3.3 : Sous-Schéma pour le QUESTIONNAIRE

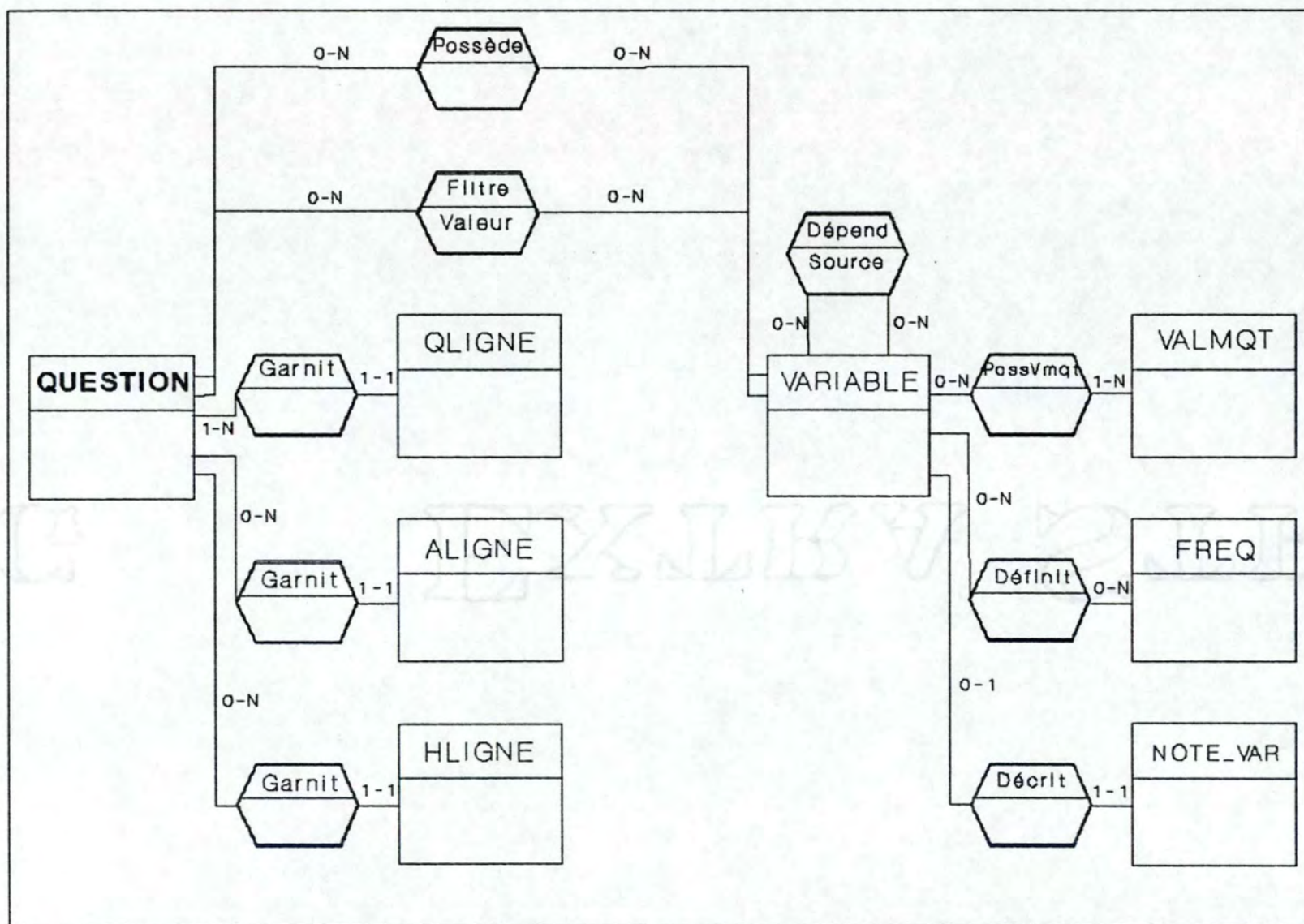


Fig. 5.3.4 : Sous-Schéma pour la QUESTION

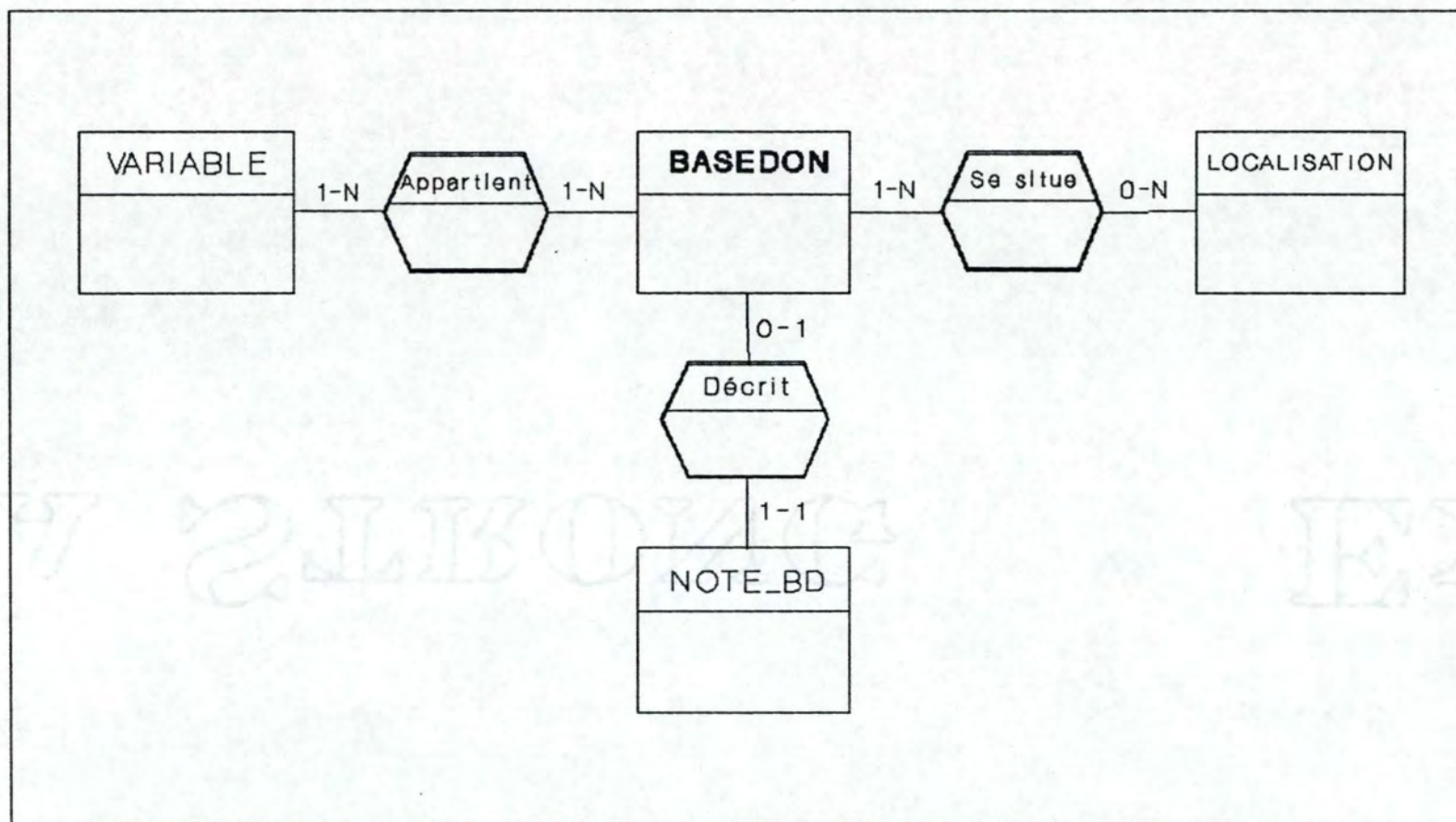


Fig. 5.3.5 : Sous-Schéma pour la BASE DE DONNEES

5.4 Le modèle MAG¹ du système d'informations

Après la description du modèle Entité/Association (E/A), nous disposons de tous les éléments nécessaires pour la présentation du modèle MAG², . Ce modèle a l'avantage de transformer le schéma E/A en une spécification plus opérationnelle de la base de données. Il est donc en quelque sorte un modèle intermédiaire entre le modèle conceptuel et celui du Système de Gestion de Bases de Données (SGBD). Dans notre cas, le SGBD sera un modèle relationnel qui convient idéalement au système de gestion de SIR.

Avant de présenter le résultat de la transformation E/A vers MAG, rappelons brièvement les concepts liés au modèle MAG.

L'ARTICLE est une unité d'information, équivalent au concept d'entité du schéma E/A. De même, le TYPE D'ARTICLE est un ensemble d'articles qui possèdent des propriétés communes et ce type d'articles correspond au type d'entité.

L'attribut du schéma E/A sera représenté par un ITEM lié au type d'article, tout comme l'était l'attribut pour le type d'entité. Il existe des items simples identifiants (pas de symboles) et des items simples non identifiants (triangle inversé).

Le type d'association sans attribut sera remplacé par un TYPE DE CHEMIN reliant les types d'article. La connectivité est substituée par les notions de CONTRAINTE D'EXISTENCE et de CLASSE FONCTIONNELLE.

Pour ce qui est de la représentation graphique, la contrainte d'existence se représente par un trait situé à proximité du type d'article et la classe fonctionnelle par un diabol (dans le cas d'une connectivité N-N), par un triangle (pour les connectivités 1-N ou N-1) ou par aucun symbole (si connectivité 1-1).

1 L'abréviation MAG signifie Modèle d'Accès Généralisé.

2 Ce modèle est décrit par J.-L. HAINAUT.

Le type d'association avec attribut sera remplacé par un type d'article auquel sera attaché les items correspondants aux attributs.

Ensuite, on construit les chemins d'accès entre les différents types d'article (aussi bien pour ceux qui représentent les entités que ceux associés aux associations). Dans le cas précis du type d'article représentant une association, le type de chemin sera dit OBLIGATOIRE.

Cette première étape de transformation terminée, il est possible de modifier à nouveau ce modèle afin de se rapprocher graduellement de la spécification SGBD. Dans cette deuxième phase, nous remplacerons les types de chemin de classe fonctionnelle N-N en deux types de chemin de classe 1-N.

Enfin, la troisième et dernière phase consiste en la transformation du modèle construit dans la deuxième phase en un modèle conforme au modèle relationnel. Ce schéma relationnel adopte certaines particularités :

- * de ne pas posséder de types chemins,
- * de n'avoir que des items simples (non répétitifs) et élémentaires (non décomposables),
- * le fait que chaque type d'article comprend au moins un item.

5.4.1 Les relations

Les contraintes référentielles

Ces contraintes sont nécessaires pour effectuer le passage du MAG vers le Relationnel.

- * LIAISON (:IDETUD1) in ETUDE (:IDETUD)
- * LIAISON (:IDETUD2) in ETUDE (:IDETUD)
- * FONDE (:IDETUD) in ETUDE (:IDETUD)
- * FONDE (:IDENQ) in ENQUETE (:IDENQ)


```

* A (:IDETUD) in ETUDE (:IDETUD)
* A (:IDRESP) in RESPONSABLE (:IDRESP)
* POSSEDE (:IDETUD) in ETUDE (:IDETUD)
* POSSEDE (:IDRESP) in RESPONSABLE (:IDRESP)
* NOTE_ENQ (:IDNOTENQ) in ENQUETE (:IDENQ)
* CONCERNE (:IDENQ) in ENQUETE (:IDENQ)
* CONCERNE (:IDPOP) in POPULATION (:IDPOP)
* EXTRAIT (:IDPOP) in POPULATION (:IDPOP)
* EXTRAIT (:IDECHANT) in ECHANTILLON (:IDECHANT)
* POST_STRATIFIE (:IDECHANT) in ECHANTILLON (:IDECHANT)
* POST_STRATIFIE (:IDCAT) in CATEGORIE (:IDCAT)
* DIVISE (:IDECHANT) in ECHANTILLON (:IDECHANT)
* DIVISE (:IDSSECH) in SOUS-ECHANTILLON (:IDSSECH)
* EST SOUS (:IDRESP) in RESPONSABLE (:IDRESP)
* EST SOUS (:IDQUAIRE) in QUESTIONNAIRE (:IDQUAIRE)
* RECOIT (:IDSSECH) in SOUS-ECHANTILLON (:IDSSECH)
* RECOIT (:IDQUAIRE) in QUESTIONNAIRE (:IDQUAIRE)
* SE DIVISE (:IDQUAIRE) in QUESTIONNAIRE (:IDQUAIRE)
* SE DIVISE (:IDPAGE) in PAGE/THEME (:IDPAGE)
* COMPORTE (:IDPAGE) in PAGE/THEME (:IDPAGE)
* COMPORTE (:IDQUEST) in QUESTION (:IDQUEST)
* QUESTION (:FORMAT) in FORMAT (:IDFORMAT)
* FORMAT (:TYPE) in TYPQUEST (:IDTYPE)
* QLIGNE (:IDQUEST) in QUESTION (:IDQUEST)
* ALIGNE (:IDQUEST) in QUESTION (:IDQUEST)
* HLIGNE (:IDQUEST) in QUESTION (:IDQUEST)
* CONTIENT (:IDQUAIRE) in QUESTIONNAIRE (:IDQUAIRE)
* CONTIENT (:IDQUEST) in QUESTION (:IDQUEST)
* POSSESSION (:IDQUEST) in QUESTION (:IDQUEST)
* POSSESSION (:IDVAR) in VARIABLE (:IDVAR)
* FILTRE (:IDVAR) in VARIABLE (:IDVAR)
* FILTRE (:IDQUEST) in QUESTION (:IDQUEST)
* NOTE_VAR (:IDNOTVAR) in VARIABLE (:IDVAR)
* DEPEND (:IDVAR1) in VARIABLE (:IDVAR)
* DEPEND (:IDVAR2) in VARIABLE (:IDVAR)
* POSSVMQT (:IDVAR) in VARIABLE (:IDVAR)
* POSSVMQT (:IDVMQT) in VALMQT (:IDVMQT)
* DEFINIT (:IDFREQ) in FREQ (:IDFRQ)
* DEFINIT (:IDVAR) in VARIABLE (:IDVAR)
* APPARTIENT (:IDVAR) in VARIABLE (:IDVAR)

```


- * APPARTIENT (:IDBD) in BASEDON (:IDBD)
- * SE SITUE (:IDBD) in BASEDON (:IDBD)
- * SE SITUE (:IDMACH) in LOCALISATION (:IDMACH)
- * NOTE_BD (:IDNOTBD) in BASEDON (:IDBD)

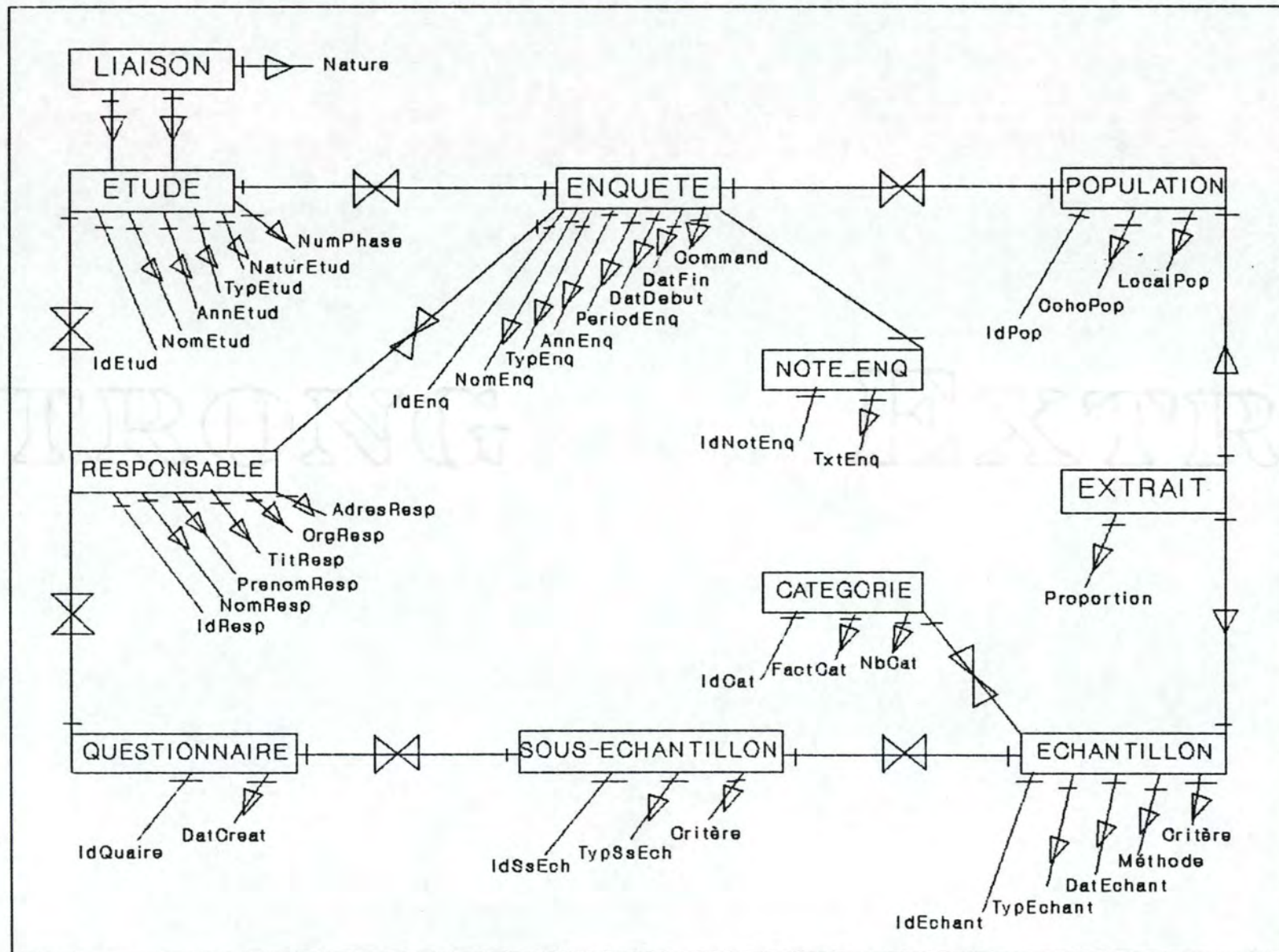


Fig. 5.4.1 : Schéma MAG pour l'ENQUETE

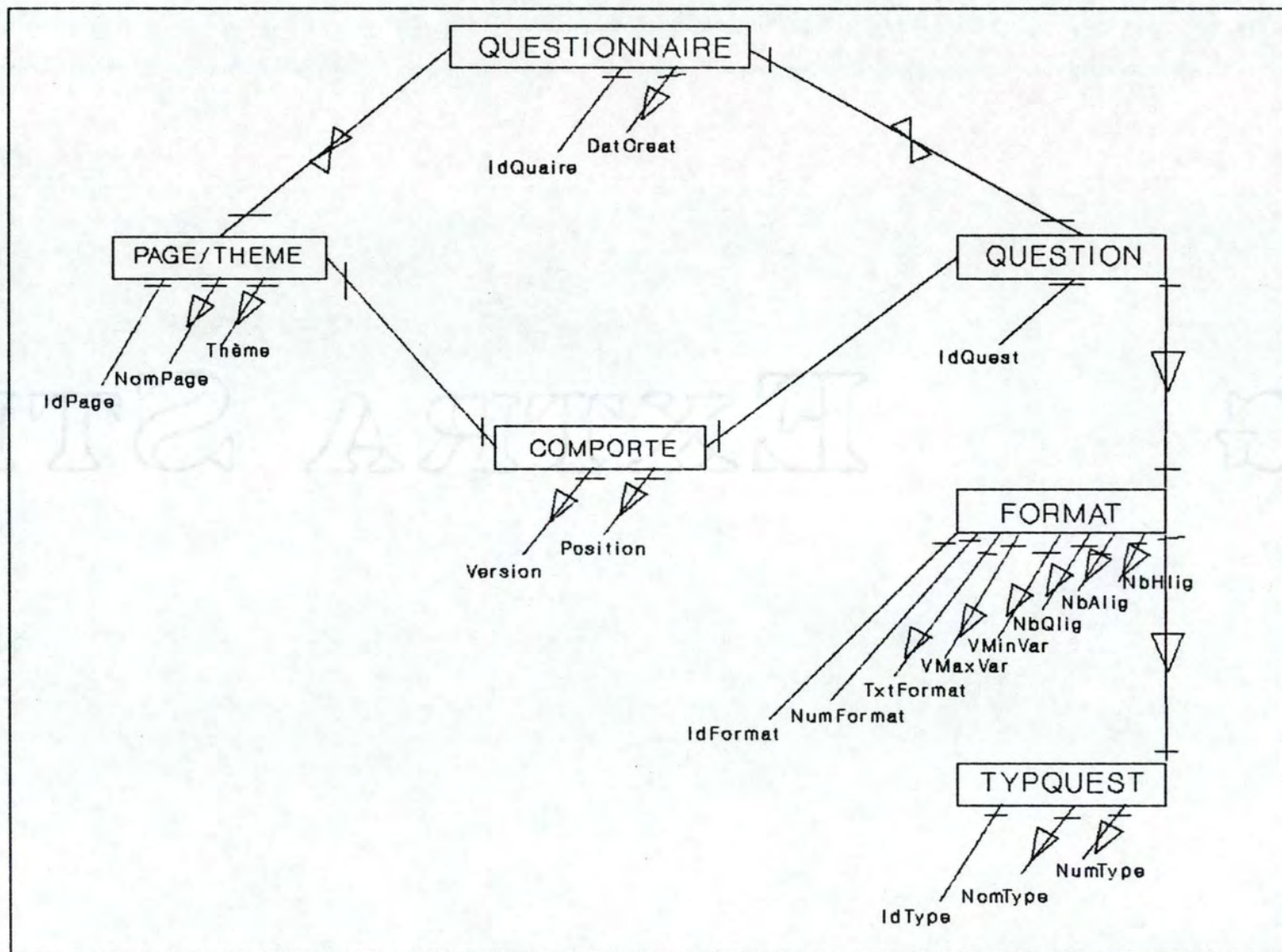


Fig. 5.4.2 : Schéma MAG pour le QUESTIONNAIRE

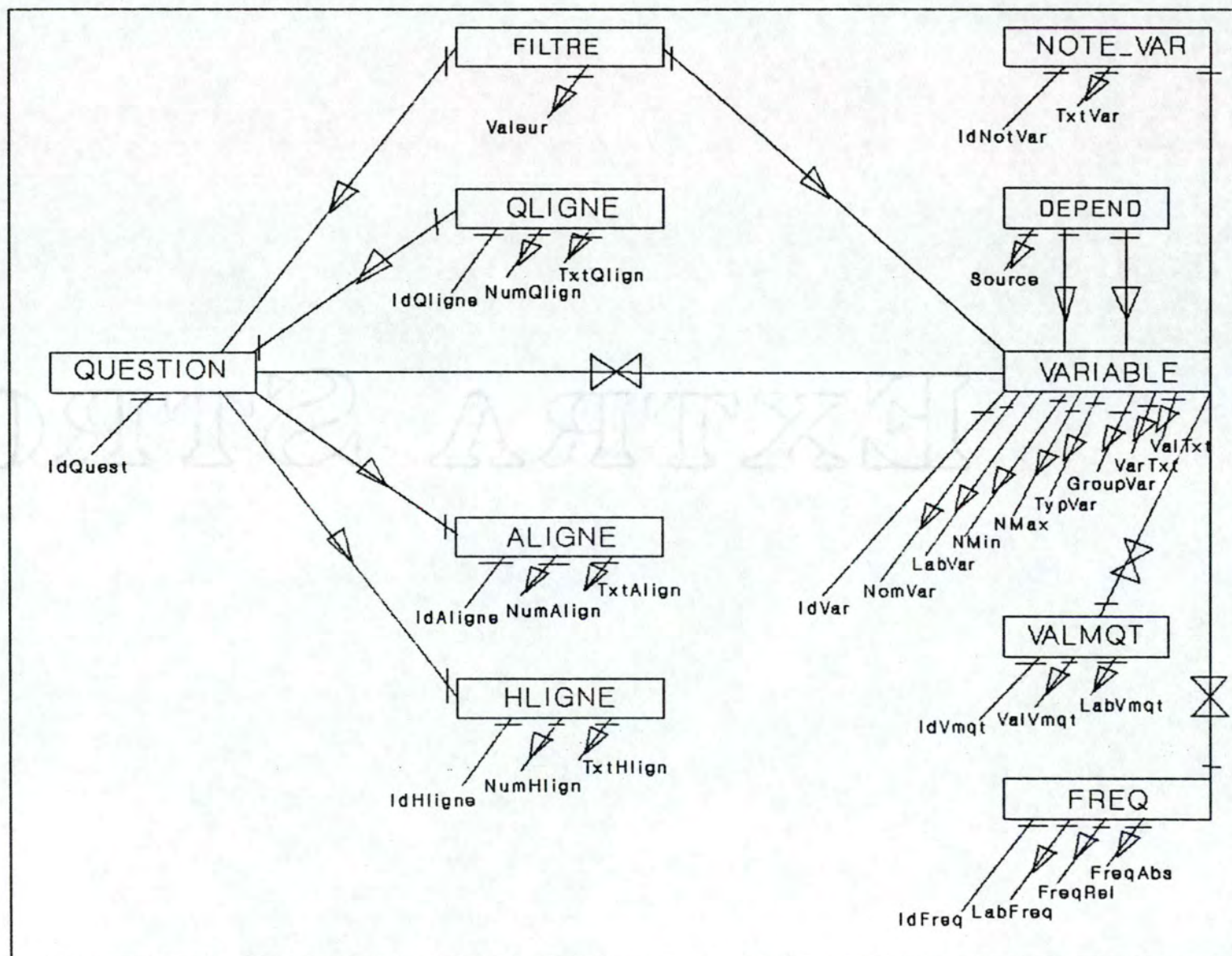


Fig. 5.4.3 : Schéma MAG pour la QUESTION

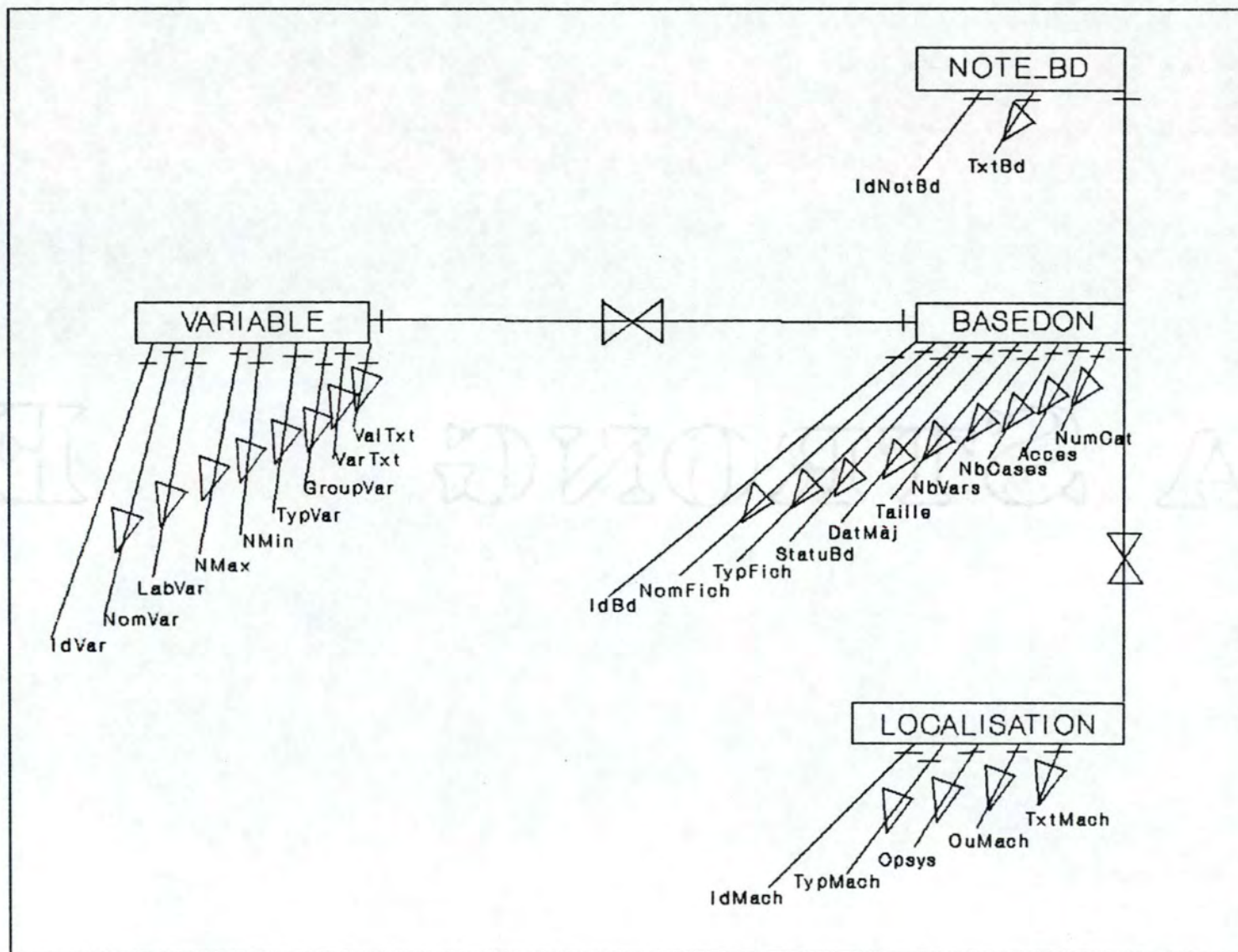


Fig. 5.4.4 : Schéma MAG pour la BASE DE DONNEES

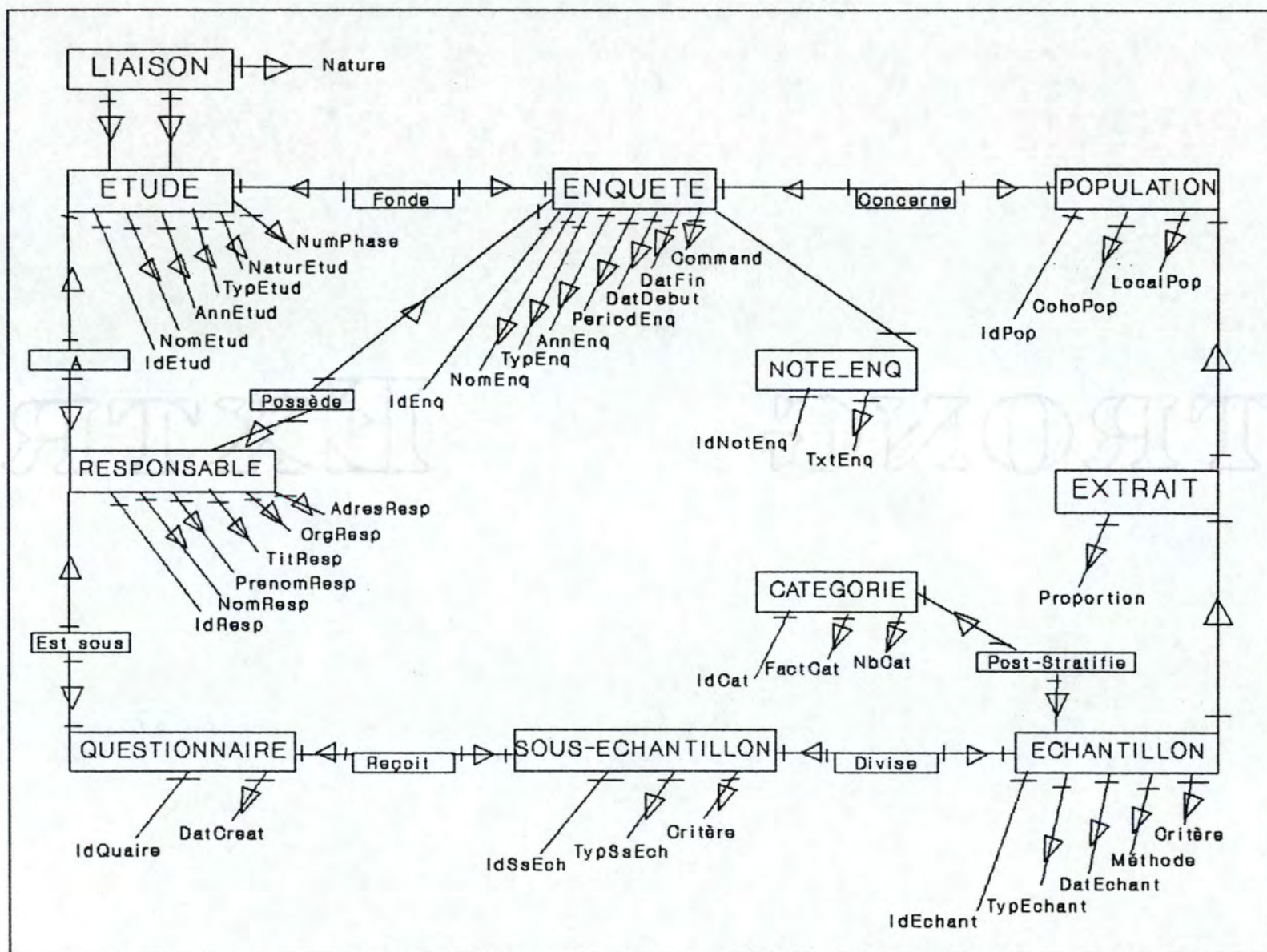


Fig. 5.4.5 : Schéma MAG transformé pour l'ENQUETE

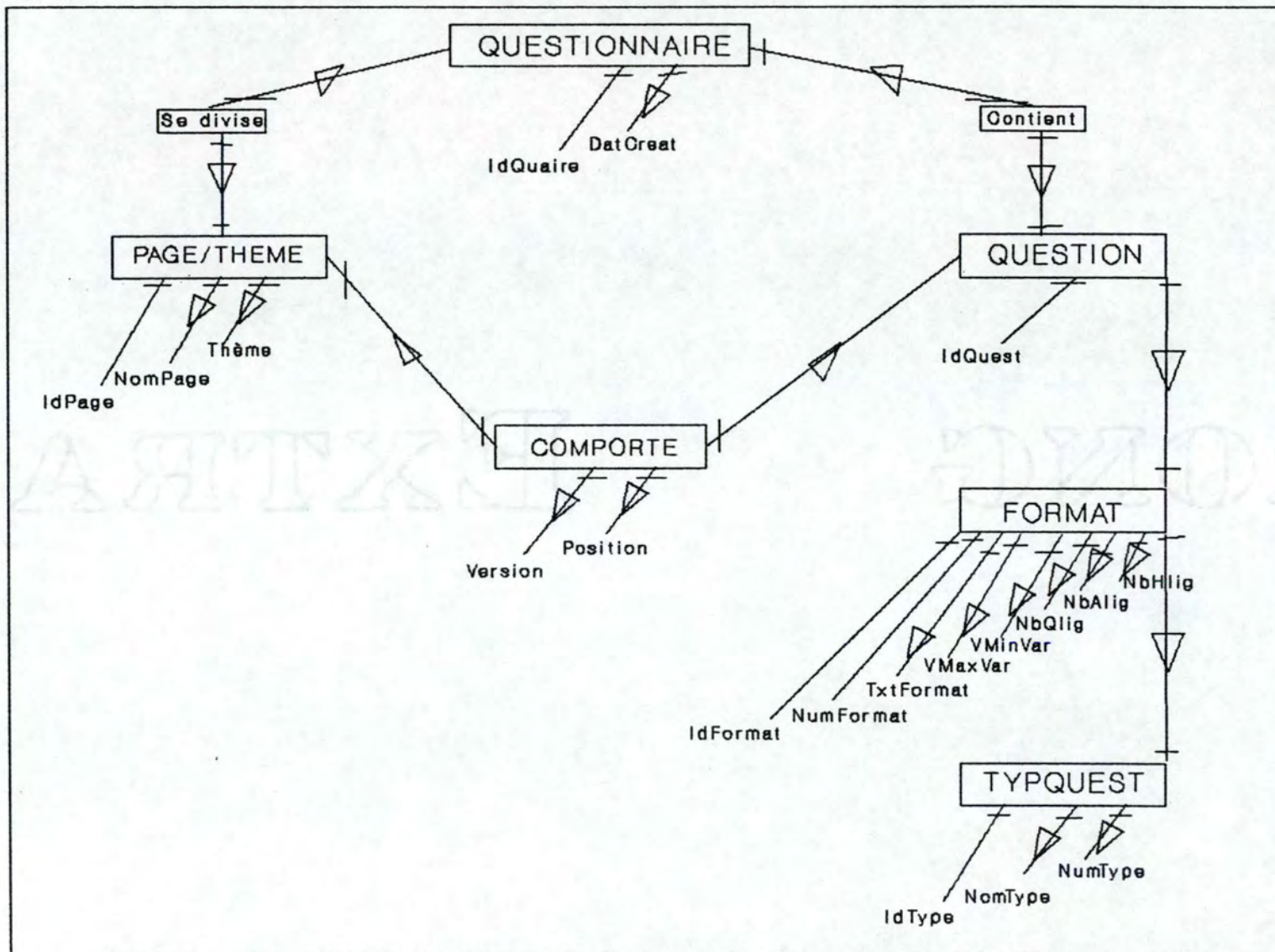


Fig. 5.4.6 : Schéma MAG transformé pour le QUESTIONNAIRE

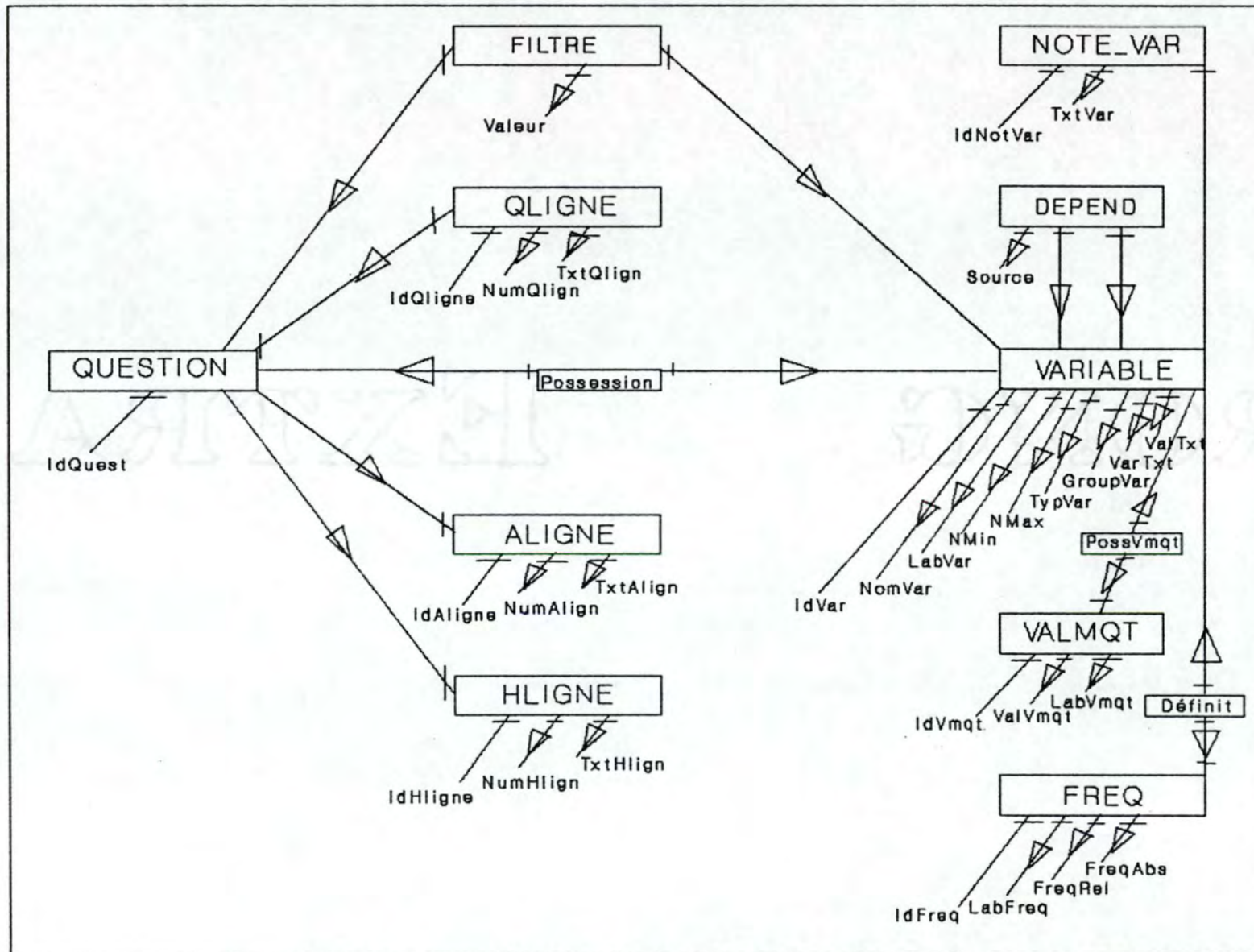


Fig. 5.4.7 : Schéma MAG transformé pour la QUESTION

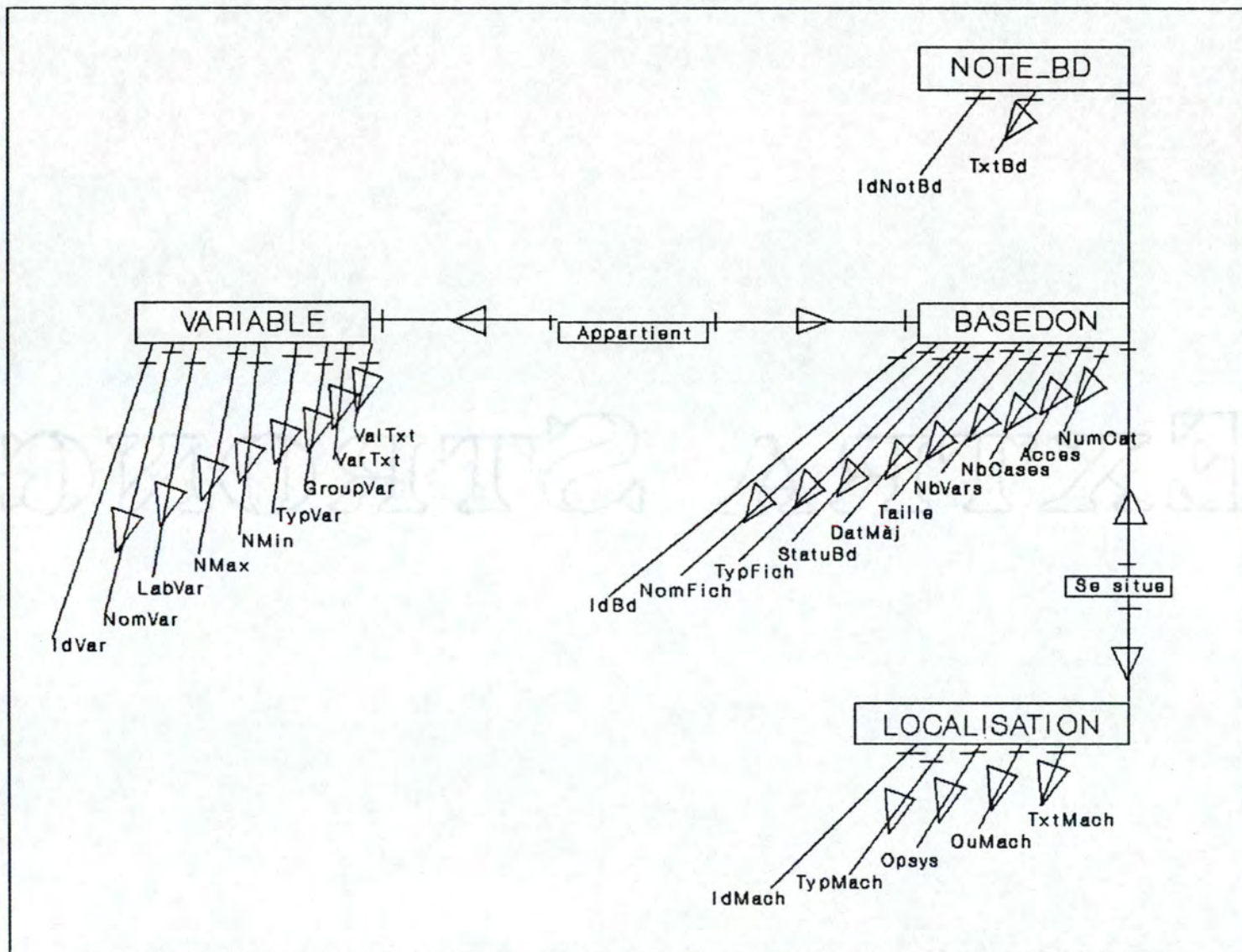


Fig. 5.4.8 : Schéma MAG transformé pour la BASE DE DONNEES

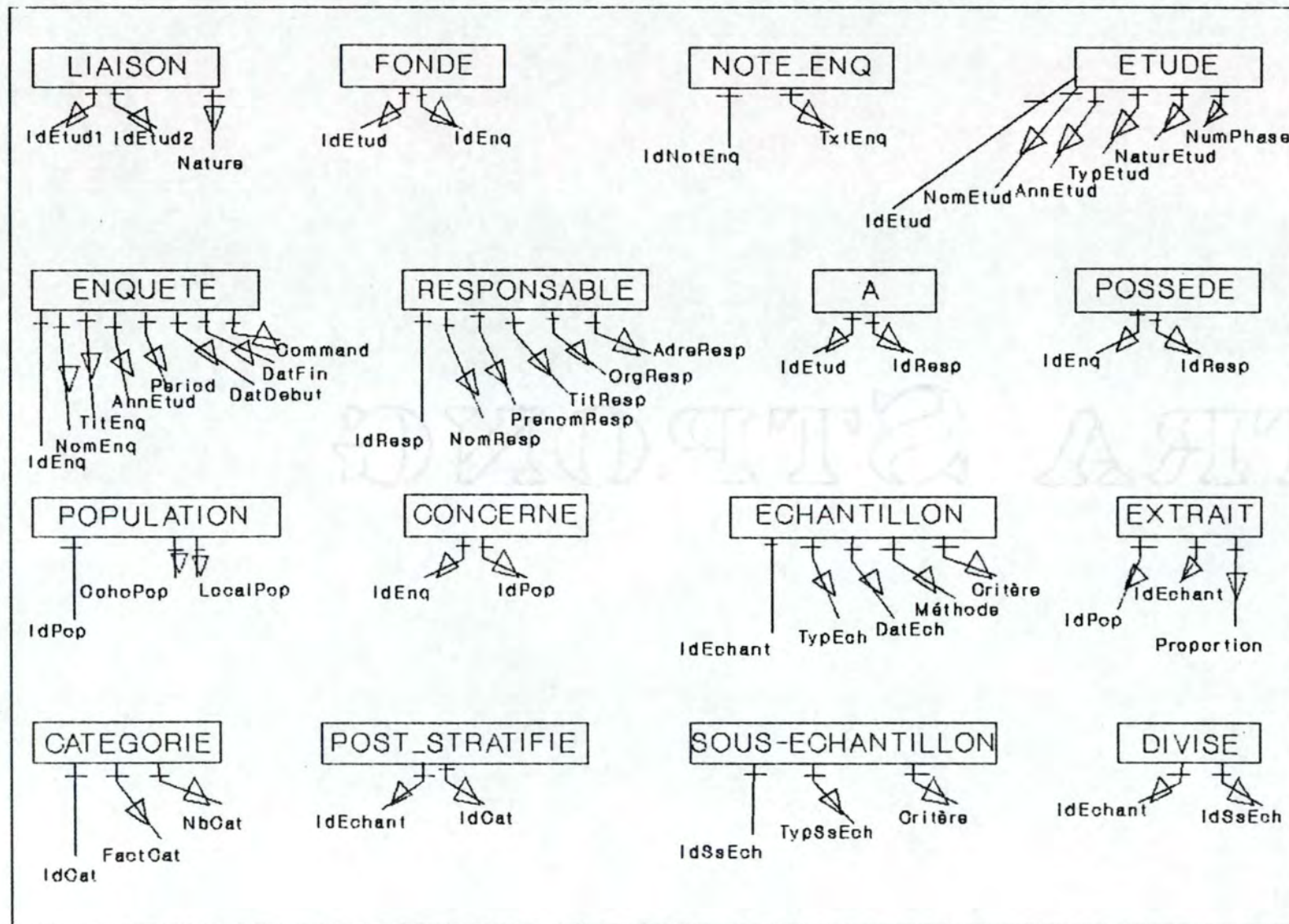


Fig. 5.4.9 : Schéma MAG - Relationnel pour l'ENQUETE

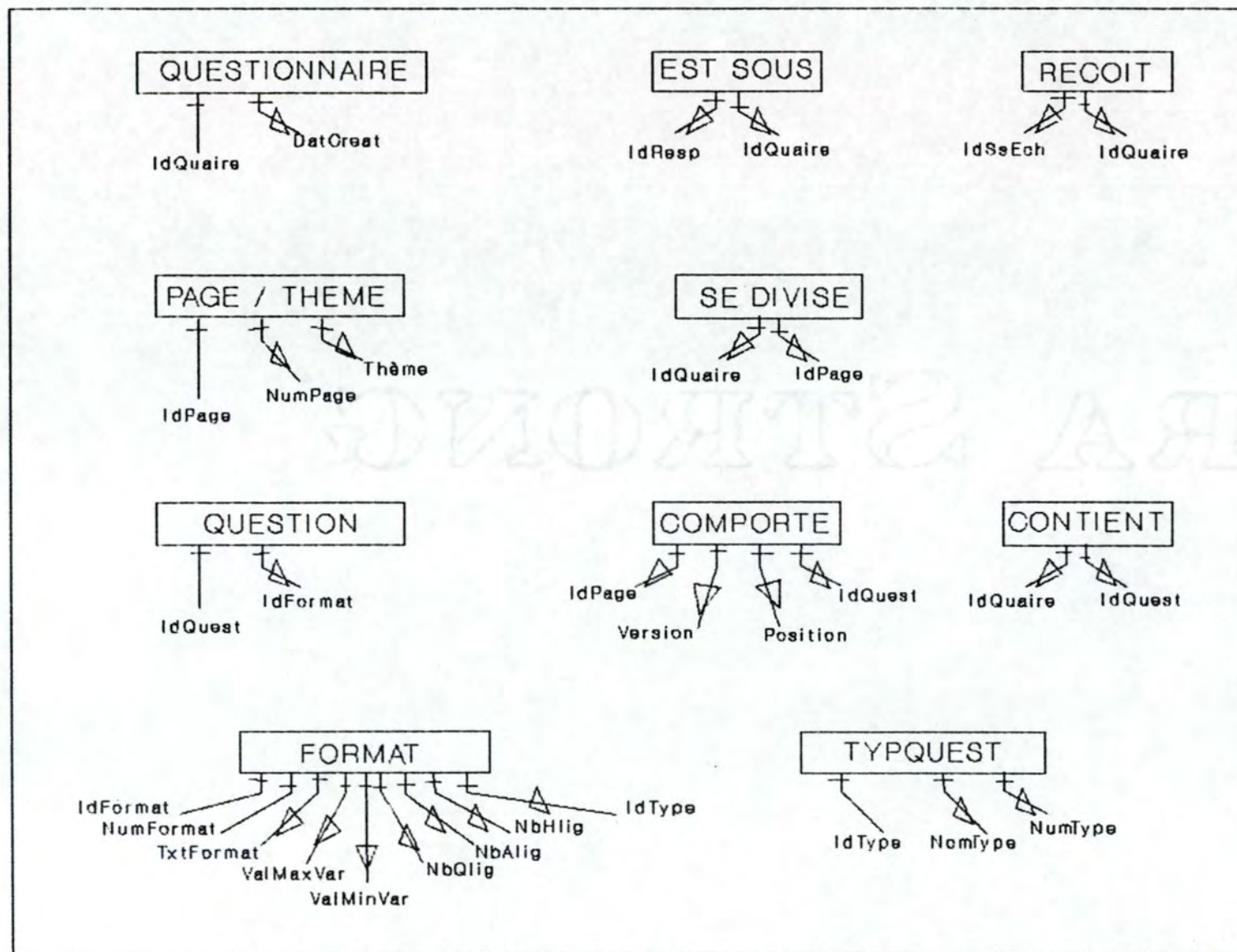


Fig. 5.4.10 : Schéma MAG - Relationnel pour le QUESTIONNAIRE

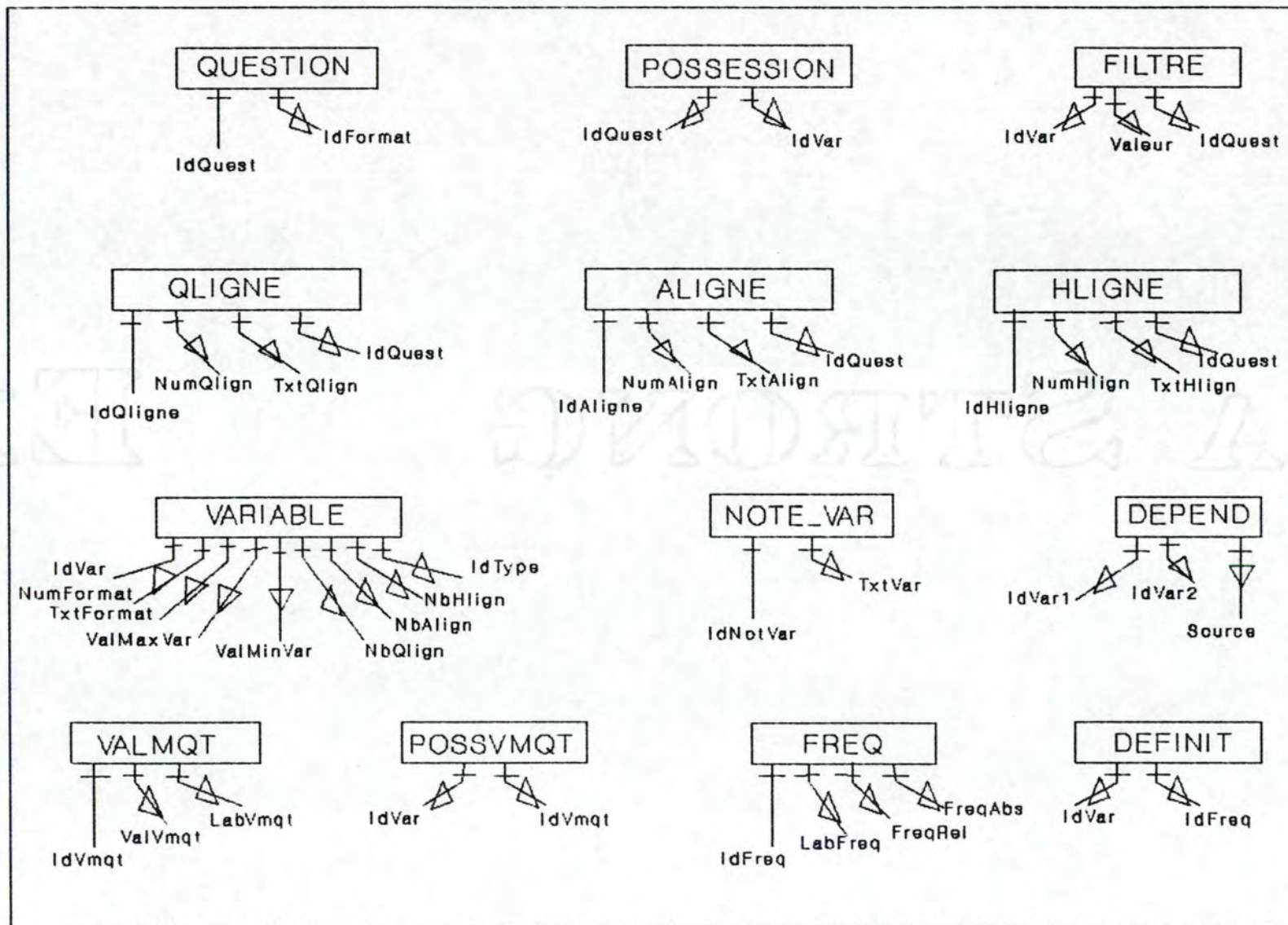


Fig. 5.4.11 : Schéma MAG - relationnel pour la QUESTION

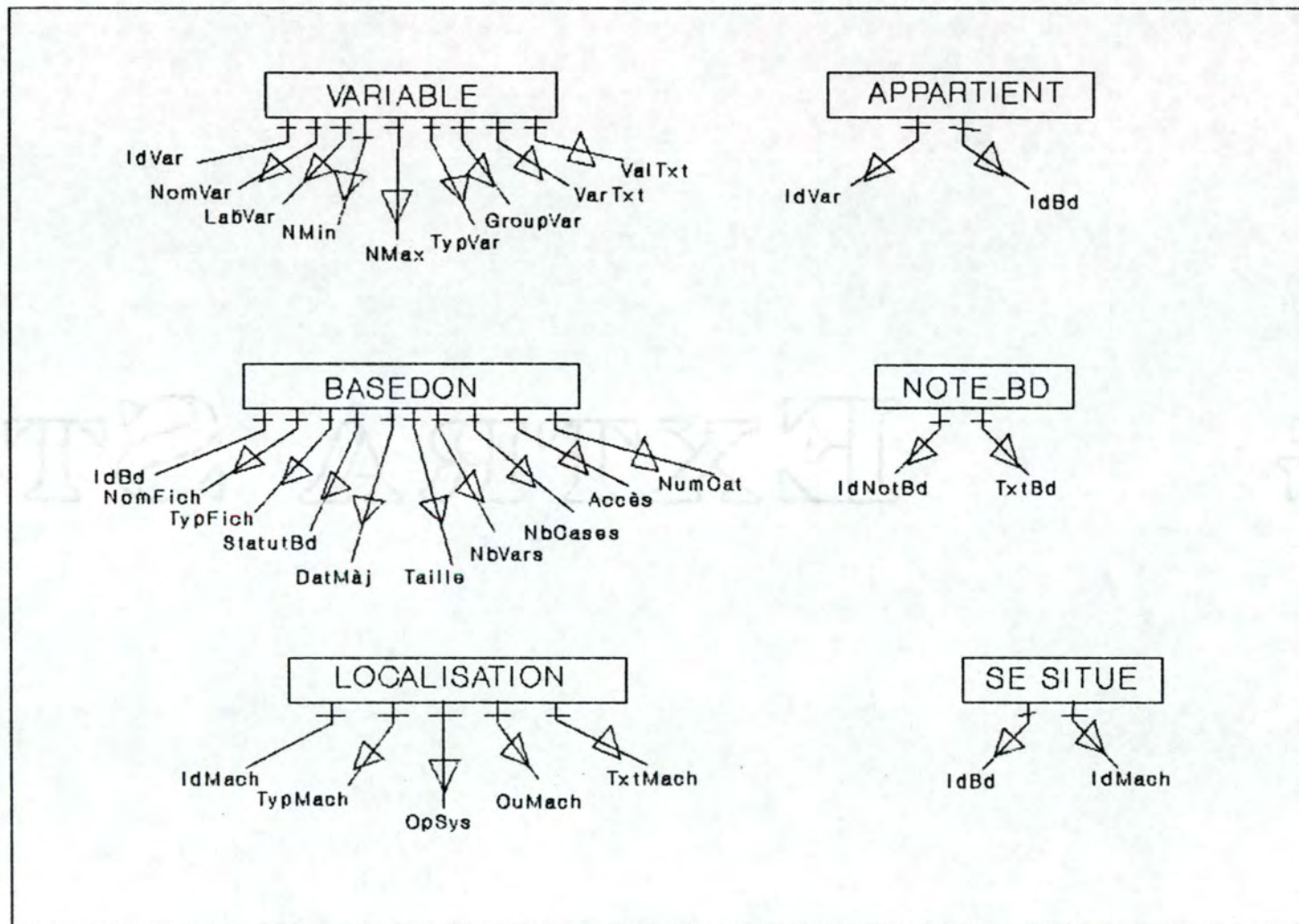


Fig. 5.4.12 : Schéma MAG - Relationnel pour la BASE DONNEES

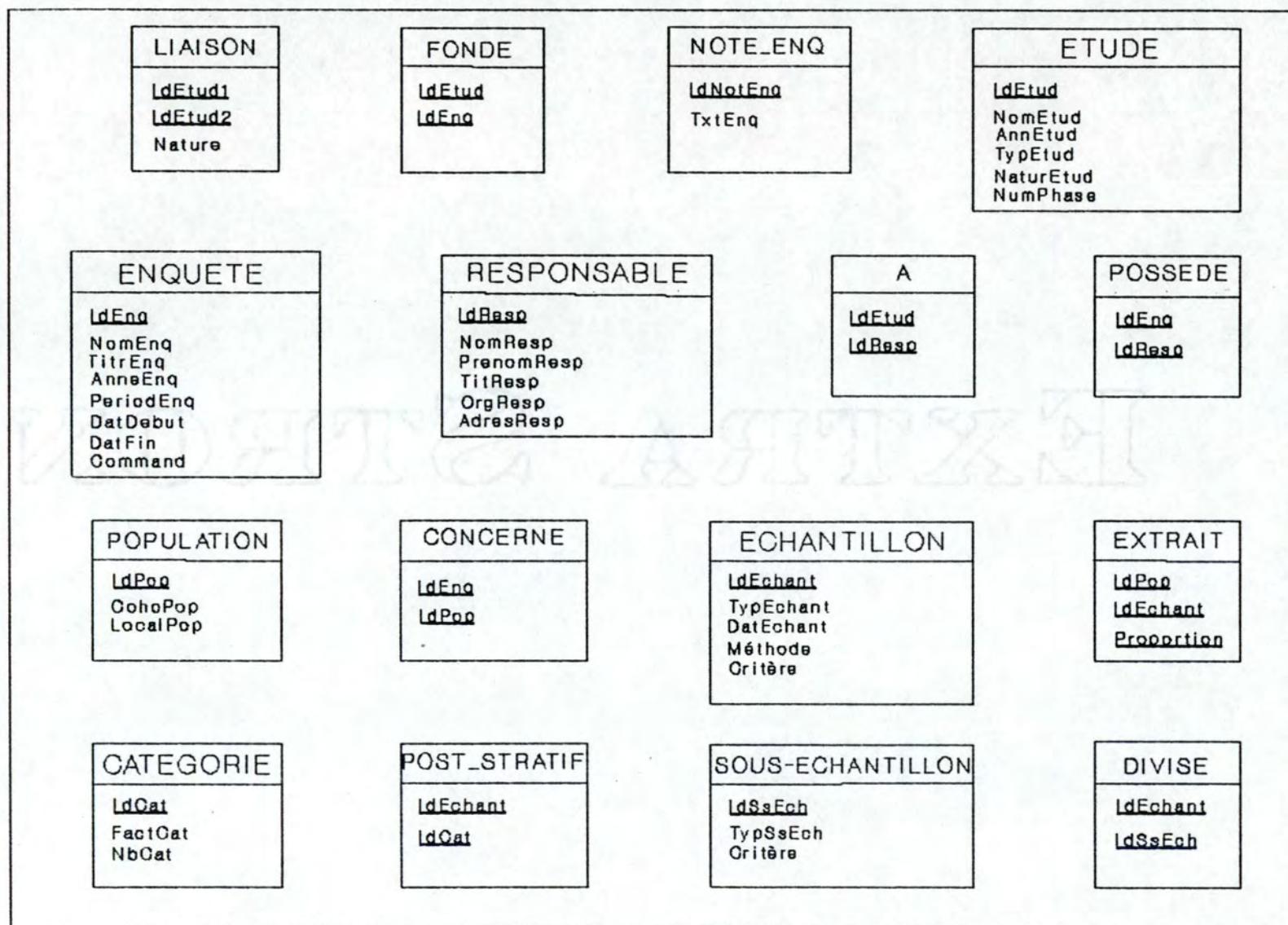


Fig. 5.4.13 : Schéma Relationnel pour l'ENQUETE

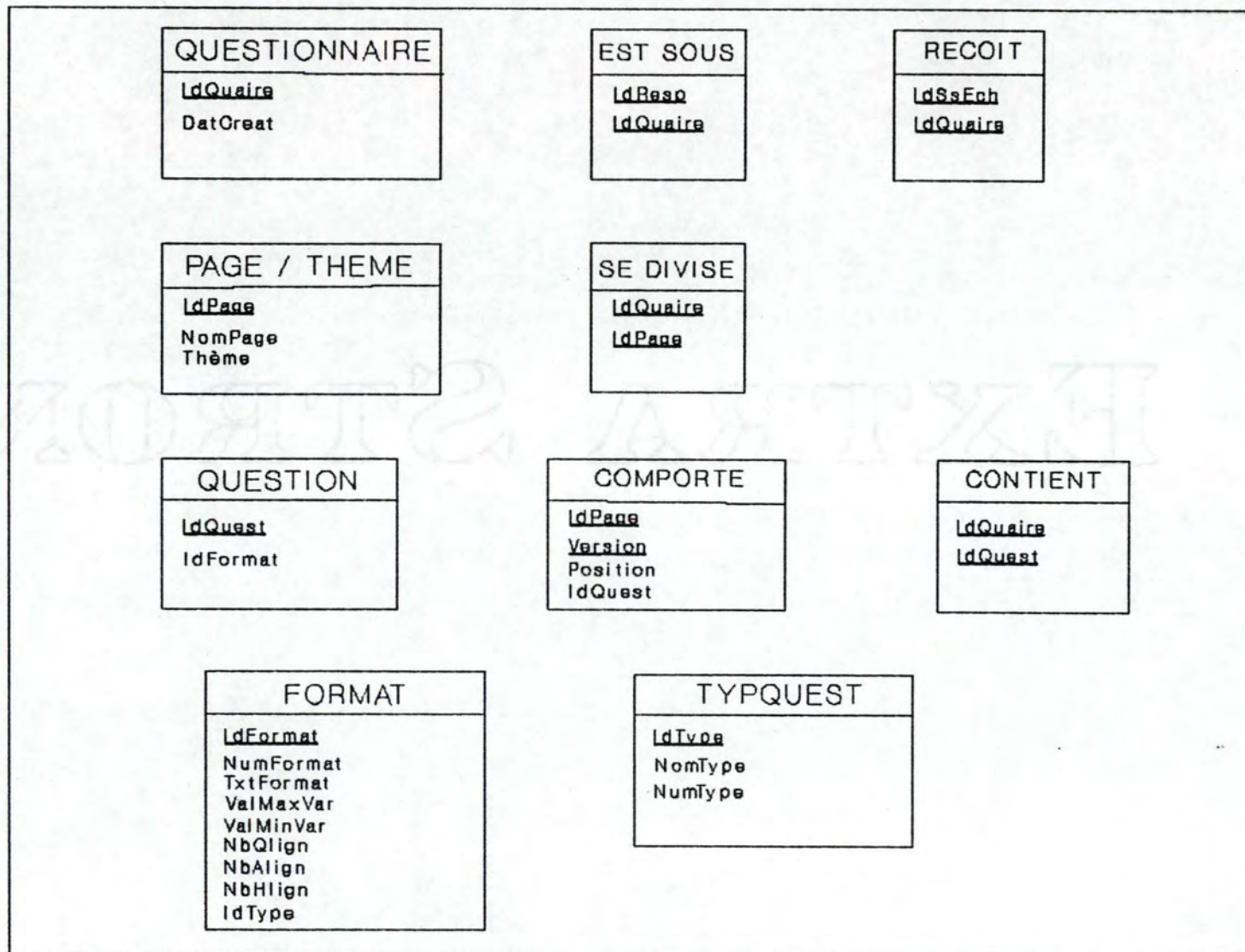


Fig. 5.4.14 : Schéma Relationnel pour le QUESTIONNAIRE

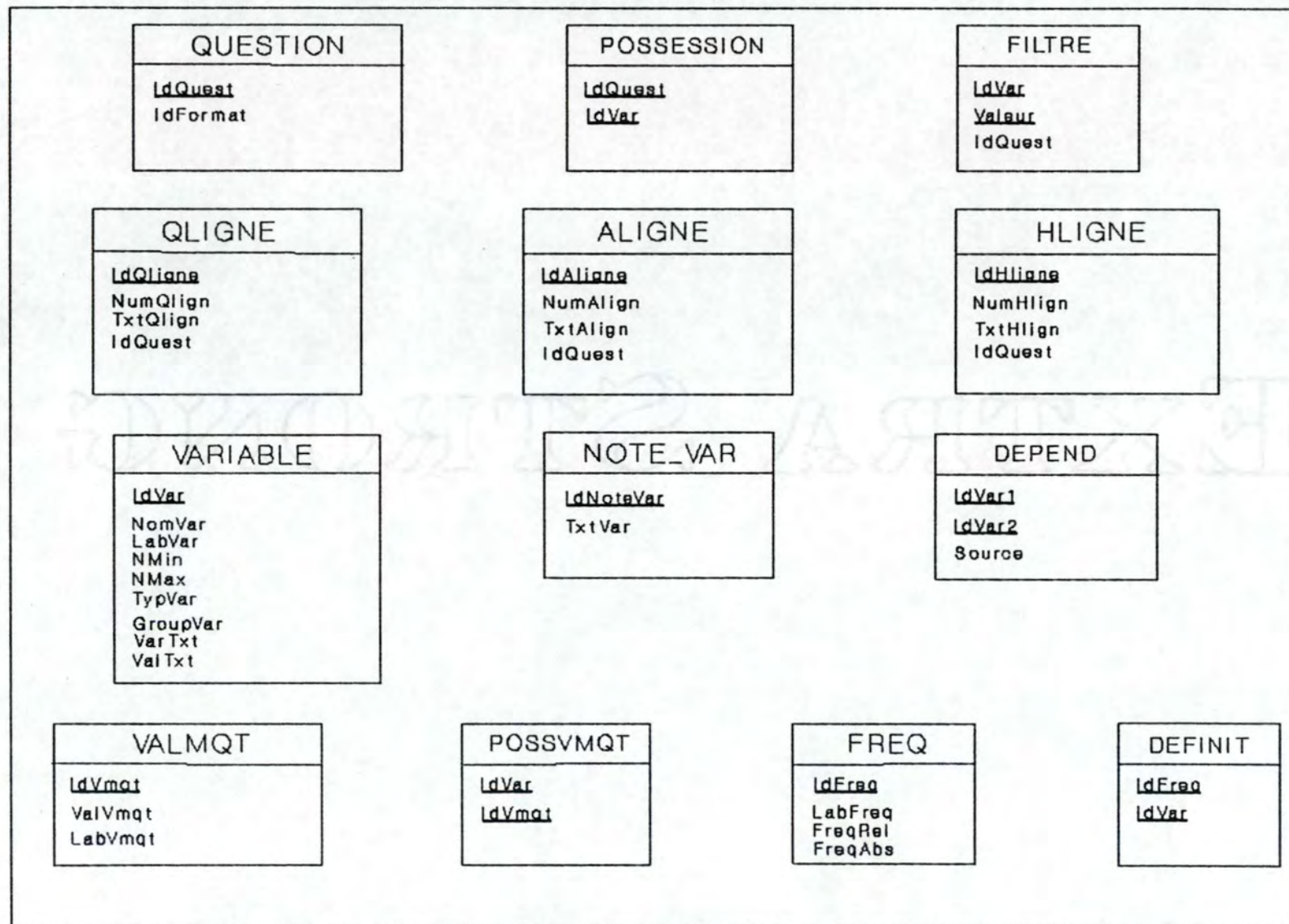


Fig. 5.4.15 : Schéma Relationnel pour la QUESTION

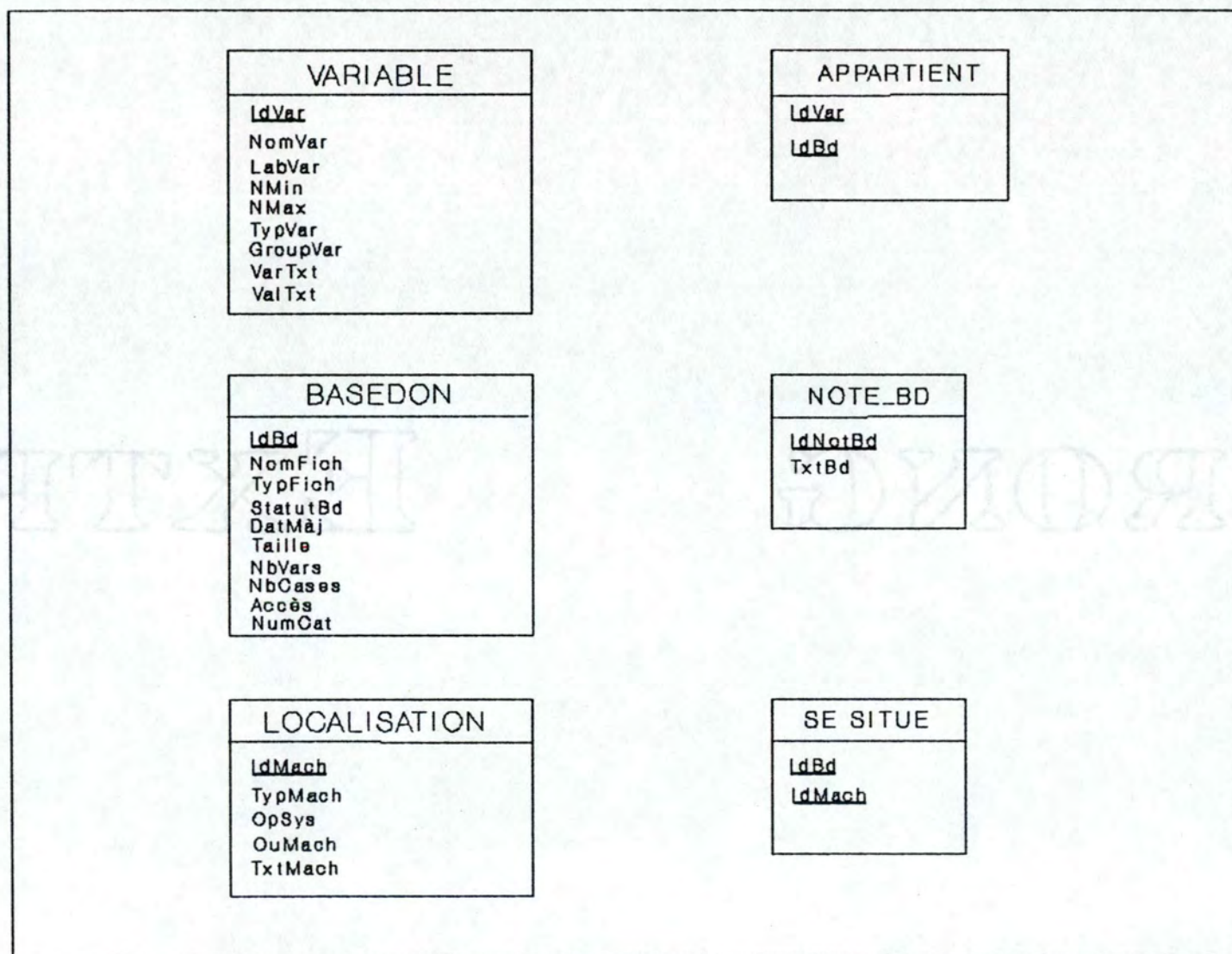


Fig. 5.4.16 : Schéma Relationnel pour la BASE DE DONNEES

6. LA DOCUMENTATION DES VARIABLES

6.1 L'intégration du programme DOCPROJ dans le projet METADATA database

Rappelez-vous la figure 2.3.1 du chapitre 2, section 2.3.1, nous vous indiquions qu'actuellement, les travaux de documentation des bases de données du CES se limitent à la conception et à l'élaboration de la base intermédiaire METADATA.

Parmi les différentes sources qui vont intervenir dans cette conception, on trouve QUESTMAST. Notre projet DOCPROJ de documentation des variables aura pour objectif d'extraire les informations du fichier principal de QUESTMAST, à savoir le MASTERFILE.

De manière plus précise, voici comment s'intègre notre travail dans l'élaboration de METADATA¹ :

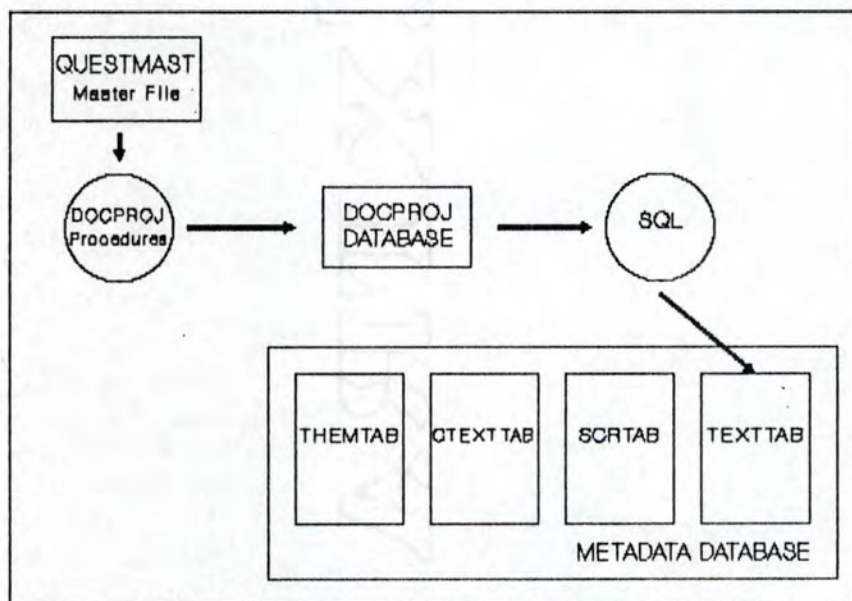


Fig. 7.1.1 : Intégration de DOCPROJ dans METADATA

1 Extrait de "Processing a Survey using Metadata Techniques", Fairgrieve J., CES, 1991.

Nous pouvons à présent passer à l'articulation du programme DOCPROJ.

6.2 La spécification du programme DOCPROJ

6.2.1 Détermination du schéma Entité/Association

Notre programme a pour objectif précis de garnir les champs VARLAB et VALLAB des variables contenues dans la base de données documentation METADATA au moyen des informations du MASTERFILE².

Le champ VARLAB contient le texte de la question posée dans le questionnaire et le champ VALLAB contient la description des réponses possibles.

Après analyse approfondie du mécanisme de création des questions dans QUESTMAST, on a pu remarquer que le numéro de format joue un rôle essentiel. En effet, c'est le numéro de format qui détermine le nombre de variables attachées à une question, le nombre de lignes de réponses (ALIGNÉ) et le nombre de lignes d'en-têtes (HLIGNE). Suivant la valeur du format, on établit de manière précise la ou les combinaisons des ALIGNÉ et HLIGNE qui vont servir à remplir ces deux champs.

Reprenons le schéma Entité/Association³ qui concerne les variables pour préciser les entités qui interviennent dans la réalisation de la documentation des variables :

2 Le Master File est un fichier de questions défini au chapitre 4, section 4.2.

3 Voir les schémas Entité/Association décrits au chapitre 5, section 5.3.

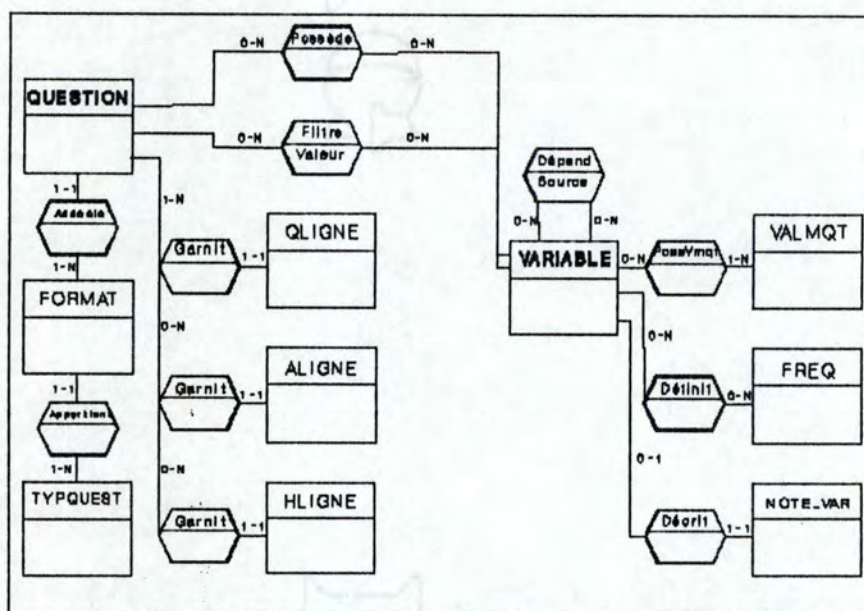


Fig. 6.2.1: Schéma de la relation QUESTION - VARIABLE

D'après ce qu'il a été dit ci-dessus, il semble évident que les entités VARIABLE, QLIGNE, ALIGNE, HLIQNE et FORMAT soient les entités principales qui jouent un rôle dans le remplissage des champs VARLAB et VALLAB.

Sous quelles formes sont présentés ces entités ? Autrement dit, de quelle forme sont les inputs de notre programme de documentation ? Nous vous proposons de répondre à ces questions dans la section suivante.

6.2.2 Implémentation de la documentation des variables

Au départ de ce projet, nous disposions de trois éléments. Le premier, et de loin le plus déterminant, était le fichier de définition des questions MASTER FILE. Le deuxième est un fichier INDEX, que QUESTMAST génère automatiquement lors de la création des questions et enfin, le troisième est un autre

fichier index QUESVAR, créé à partir des données mêmes de la base de données documentation METADATA.

Avant de présenter la méthodologie de développement de notre programme, précisons que toutes les bases de données du CES sont construites sur SIR (Statistical Information Retrieval) qui se base lui-même sur le système de gestion de bases de données DBMS (Data Base Management System) dont un des langages est PQL (Procedural Query Language).

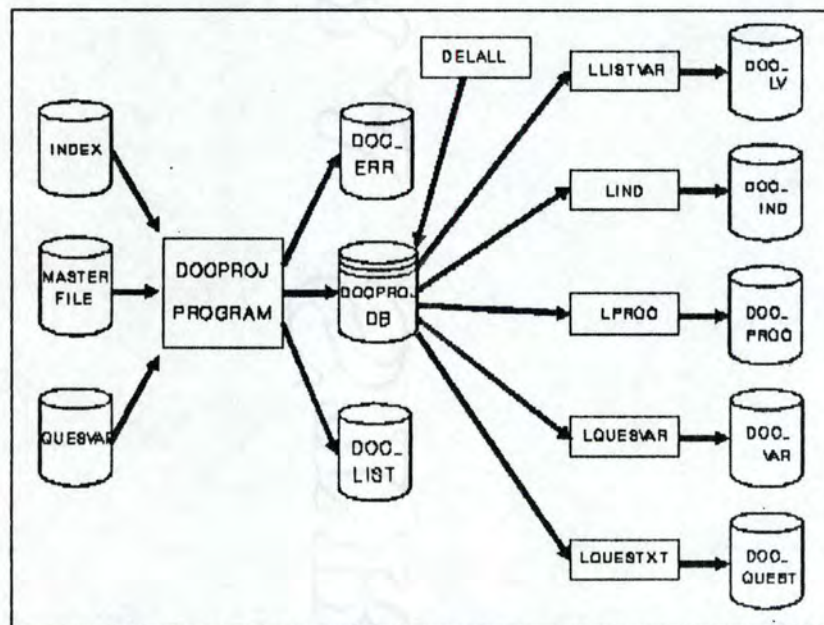


Fig. 6.2.2 : Les inputs et outputs du programme DOOPROJ

Après exécution du programme DOCPROJ⁴, trois outputs sont produits. Le premier est la mise à jour de notre base de données documentation DOCPROJ qui est une base de données intermédiaire stockant les données avant que celles-ci soient définitivement transférées dans la base de données documentation METADATA. Le deuxième output est le fichier DOC_ERR qui contient les erreurs de syntaxe ou de traitement qui sont apparus lors de l'exécution. Le troisième et dernier output est le fichier DOC_LIST qui contient toutes les données

4 Les programmes PQL de DOCPROJ se trouvent en Annexe 5.

extraites du MASTER FILE et qui figurent également dans DOCPROJ.

Ainsi, sont stockées dans notre base de données DOCPROJ, les données ne faisant pas l'objet d'erreurs détectées à l'exécution. Enfin, par souci de facilité de consultation de notre base de données DOCPROJ, on a construit différents petits programmes permettant de sauver dans un fichier de type texte toutes les valeurs des records de DOCPROJ.

Voici un exemple du fonctionnement du programme.

Exemple

Soit le MASTERFILE suivant⁵ :

```
D50 .      22
Q~5Did you think about applying for any courses at college $or
university,
  before last November (1988)?~
  $<include part-time courses>
  $<tick one box>
HYES
HNO
A<if NO, please go to the next page>
Q~5Did you apply for any courses at college or university,
  $before last November (1986)?~
  $<include part-time courses>
  $<tick one box>
HYES
HNO
A<if NO, please go to question 4>
C From SPRING87
VCOLS=1      NAME=NOTSINCE      RANGE=1-2

D95 .H      90
Q~5Write in the names of the subjects you studied for $Sixth
Year Studies (CSYS),
  [tick] the box if you sat the exam and [ring] the result you
got
  for each one~
  <(n = no award)>
  $<if none, write 'none'>
H~5CSYS~
A.....
A.....
A.....
A.....
C From SPRING87
VCOLS=4111*5  NAME=SYSSUBJ1-SYSGRAD5RANGE=1-7
```

5 Les questions D50. et D151. correspondent respectivement aux questions n°1 de la page 7 et n°1 de la page 8 de l'Annexe 1.


```

D151.      77
Q~5Over the next few years, do you hope to...~
$<don't count anything you are already doing>
$<tick one box for each line>
H[yes]
H[no]
H[not sure]
A...start a degree course at university, or somewhere else?
A...start another full-time college course?
A...start on the Youth Training Scheme (YTS)?
A...start some other type of training for a job?
C From SPRING87
VCOLS=1*4      NAME=VHPEFEW1-VHPEFEW4RANGE=1-3

```

et le fichier QUESVAR,

NumQuest	NumVar	NomVar	NumFormat	GroupVar
50	1	NOTSINCV	22	NOTSINCE
95	1	SYSSUBJV	90	SYSSUBJ1-SYSGRAD5
95	2	SYSSSTUDV	90	SYSSUBJ1-SYSGRAD5
95	4	SYSGRADV	90	SYSSUBJ1-SYSGRAD5
151	1	HPEFEW1V	77	VHPEFEW1-VHPEFEW4
151	2	HPEFEW2V	77	VHPEFEW1-VHPEFEW4
151	3	HPEFEW3V	77	VHPEFEW1-VHPEFEW4
151	4	HPEFEW4V	77	VHPEFEW1-VHPEFEW4

Après exécution du programme DOCPROJ, le fichier DOC_LIST renvoie les valeurs suivantes :

LEGENDE DES VALEURS TELLES QU'ELLES APPARAISSENT DANS LE FICHIER DOC_LIST

NumQuest NumVar NomVar NumFormat NbWording

VARTEXT

VALTEXT

50 1 NOTSINCV 22 2

Did you think about applying for any courses at college\$or university,~before la
st November (1988)?~\$include part-time courses~\$tick one
box~>Did you apply for

any courses at college or university,~\$before last November
(1986)?~\$include par
t-time courses~\$tick one box

*
*

YES~~NO~>YES~~NO~~~CODER IMPLIED YES

*
*

95 1 SYSSUBJV 90 1

Write in the names of the subjects you studied for\$Sixth Year
Studies (CSYS)~tic
k the box if you sat the exam and ring the result you got~for
each one~(n = no a
ward)~\$if none, write
'none'~~~>.....

.....
*
*

TICKED~~NOT TICKED

*
*

95 2 SYSSTUDV 90 1

Write in the names of the subjects you studied for\$Sixth Year
Studies (CSYS)~tic
k the box if you sat the exam and ring the result you got~for
each one~(n = no a
ward)~\$if none, write 'none'~~~>.....

TICKED~~NOT TICKED

*
*

95 4 SYSGRADV 90 1

Write in the names of the subjects you studied for\$Sixth Year
Studies (CSYS)~tic
k the box if you sat the exam and ring the result you got~for
each one~(n = no a
ward)~\$if none, write 'none'~~~>.....

TICKED~~NOT TICKED

*
*

151 1 HPEFEW1V 77 1

Over the next few years, do you hope to...~\$don't count
anything you are already
doing~\$tick one box for each line~~~...start a degree course
at university, or
somewhere else?

yes~~no~~not sure

*
*

151 2 HPEFEW2V 77 1

Over the next few years, do you hope to...~\$don't count
anything you are already
doing~\$tick one box for each line~~~...start another full-
time college course?

*

yes~~no~~not sure

*
*

151 3 HPEFEW3V 77 1

Over the next few years, do you hope to...~\$don't count
anything you are already
doing~\$tick one box for each line~~~...start on the Youth
Training Scheme (YTS)
?

yes~~no~~not sure

*
*

151 4 HPEFEW4V 77 1

Over the next few years, do you hope to...~\$don't count
anything you are already
doing~\$tick one box for each line~~~...start some other type
of training for a
job?

yes~~no~~not sure

*
*

7. CONCLUSION

Le Centre for Educational Sociology (CES) de l'Université d'Edimbourg réalise diverses enquêtes sur l'éducation qui sont effectuées depuis 10 ans et qui ont permis de rassembler une multitude de données relatives à ce domaine de recherche.

L'objet du mémoire a été de donner une première description précise de ce centre et du travail qu'il réalise. Cette partie du mémoire peut paraître fastidieuse au lecteur, mais elle a l'avantage de mettre au clair toutes les notions utiles à la compréhension du projet de documentation, thème central du travail effectué.

En effet, étant donné l'ampleur des informations disponibles actuellement dans les bases de données du CES, un système qui permette une consultation aisée et efficace de ces bases s'avère nécessaire. DOCDB est un projet dont l'objectif est de documenter toutes les données relatives à un processus d'enquête. Il se compose de deux phases : la phase de remplissage et la phase de consultation.

Notre contribution à ce projet a consisté en la définition des schémas Entité/Association. Ceux-ci sont une proposition concrète à la conception du schéma de DOCDB qui a permis aux responsables du CES de préciser le schéma théorique de cette base documentation qui existait déjà, mais qui présentait certaines lacunes dont des problèmes de redondance et de relations entre entités.

La structure définitive de DOCDB sera acquise lorsque seront déterminées les fonctions de consultation requises par les chercheurs, ainsi que les possibilités d'accès à la base documentation. Cet aspect du problème reste encore à traiter. Nous espérons que ce problème pourra être résolu au sein même du CES ou peut-être encore par la réalisation d'un autre mémoire au cours des années à venir, notre travail ayant eu le mérite de cerner et de spécifier le cadre d'analyse et le concept-clé des fonctions de consultation, à savoir la question.

A N N E X E S

WHAT ARE YOU
DOING NOW

?

QUESTIONS FOR
YOUNG PEOPLE
IN SCOTLAND

SCOTTISH YOUNG
PEOPLE'S SURVEY

89

PERSONAL AND CONFIDENTIAL

How to answer

These questions are about the things you have been doing over the past year or so. We'd like to find out about your experiences and to hear your views. Because we are asking a large number of people, it is not possible to speak to everyone personally.

We have tried to make these questions easy and quick for you to answer. For a few questions we ask you to write in your answer.

For most of the questions there is a list of possible answers with a box printed beside each one. Please choose your answer and put a tick in the box beside it.

For example, if the answers to a question are 'yes' or 'no' and you want to say 'no' tick the box like this:

yes.	<input type="checkbox"/>	1
no	<input checked="" type="checkbox"/>	2

If none of the answers exactly fits what you want to say, choose the answer that comes nearest. If you want, you can write comments beside the question, but please always tick the answer as well.

Thank you for your help in answering these questions. We hope they let you express your own views. If you would like to tell us more in your own words, please write on the inside of the back cover.

1. Are you now...
tick one box

...on the Youth Training Scheme (YTS)?	<input type="checkbox"/> 1
...in a full-time paid job, but not on YTS?	<input type="checkbox"/> 2
...unemployed and looking for work?	<input type="checkbox"/> 3
...in full-time education at school?	<input type="checkbox"/> 4
...in full-time education at college or university?	<input type="checkbox"/> 5
...doing something else?	<input type="checkbox"/> 6

2. What have you been doing over the past few months?
tick one box for each month to say what you were doing for
all or most of the month

	1988					1989	
	Aug.	Sept.	Oct.	Nov.	Dec.	Jan.	Feb.
on the Youth Training Scheme (YTS)	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1
in a full-time paid job, but not on YTS	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2
unemployed and looking for work.	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3
in full-time education at school.	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4
in full-time education at college or university	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5
doing something else.	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6

3. Where are you staying now?
if a student, just say where you stay during term

name of town or village.....
name of Region or County.....

4. Compared with other parts of Scotland, are the job opportunities where you are staying now...
tick one box

...better than average? .	<input type="checkbox"/> 1
...about average?	<input type="checkbox"/> 2
...worse than average? .	<input type="checkbox"/> 3

5. If you had to move to another area soon, to get a job or to go to college, how much would each of these things worry you?
tick one box for each line

	a lot	a little	not at all
leaving your friends.	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
leaving your family	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
leaving your boyfriend, girlfriend or fiancé.	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
the family responsibilities you have.	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
your health	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
finding somewhere to live	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
any problems you have with money.	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3

1. Which subjects did you do in third or fourth year?
tick the boxes for third year and for fourth year

English	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	Food and Nutrition (or Cooking) . .	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
Arithmetic	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	Technical Drawing	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
Maths	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	Craft and Design	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
Physical Education (PE)	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	Woodwork	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
Religious Education (RE)	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	Metalwork	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
Geography	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	Accounting	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
History	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	other subjects		
Chemistry	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	(please write them in)		
Biology	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
Physics	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
Science	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
French	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
Art and Design (or Art)	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
Secretarial Studies (or Typing) . .	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	3rd <input type="checkbox"/>	4th <input type="checkbox"/>
Modern Studies	3rd <input type="checkbox"/>	4th <input type="checkbox"/>	3rd <input type="checkbox"/>	4th <input type="checkbox"/>

Check that you've put in all the subjects you did,
even the ones you dropped before the end of fourth year

2. Did you study for any exams in your third or fourth year?
tick one box

if YES: YES ☐ 1

NO ☐ 2 → if NO, please go to
page 4

3. In some schools, pupils take a new sort of exam instead of O grade or CSE. It is called Standard Grade. There are three types of Standard Grade exam, called Foundation, General and Credit.

Did you study any subjects for Standard Grade in third or fourth year?
tick one box

if YES: YES ☐ 1

NO ☐ 2 → if NO, please go to
the next page

4. In third or fourth year,
which subjects did you study for Standard Grade?
tick the subjects you studied for Standard Grade,
tick to say if you sat the summer exam and
ring the overall grade you got for each subject, if any

eg Latin	studied <input checked="" type="checkbox"/>	sat exam <input checked="" type="checkbox"/>	1	2	3	4	5	6	7	no award
English.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1	2	3	4	5	6	7	no award
Maths.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1	2	3	4	5	6	7	no award
Science.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1	2	3	4	5	6	7	no award
Soc & Voc.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1	2	3	4	5	6	7	no award
Art & Design.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1	2	3	4	5	6	7	no award
Computer Studies.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1	2	3	4	5	6	7	no award
Social Studies.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1	2	3	4	5	6	7	no award
Craft & Design.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1	2	3	4	5	6	7	no award
French.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1	2	3	4	5	6	7	no award
Home Economics.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1	2	3	4	5	6	7	no award
Latin.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1	2	3	4	5	6	7	no award

5. How many Standard Grade exams did you sit altogether,
in your third and fourth years?
if none, write '0'

.....Standard
.....Grades

1. Did you study any subjects for 0 grade in third or fourth year?

tick one box

YES ☐ ₁

NO ☐ ₂ → if NO, please go to question 4

if YES:

2. In third or fourth year, which subjects did you study for 0 grade?
tick the subjects you studied for 0 grade
tick to say if you sat the summer exam and then
ring the grade you got for each subject, if any

eg Latin.....	studied <input checked="" type="checkbox"/>	sat exam <input checked="" type="checkbox"/>	1 2 3 ④ 5	no award
English.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
Arithmetic.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
Maths.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
Geography.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
History.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
Chemistry.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
Biology.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
Physics.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
French.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award

other subjects
 please write them in

.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award
.....	studied <input type="checkbox"/>	sat exam <input type="checkbox"/>	1 2 3 4 5	no award

3. How many 0 grade exams did you sit altogether, in your third and fourth years?
 if none, write '0'

.....0 grades

4. While you were in third or fourth year, did you study for any other exams such as CSEs (Certificate of Secondary Education), SCOTVEC modules, RSAs, Pitmans or City & Guilds (CGLI)?
 please don't count your 0 grades or Standard Grades here
 tick one box

YES ☐ ₁

NO ☐ ₂ → if NO, please go to the next page

if YES:

Please write down the name of each subject,
 the qualification you studied for
 and the result of any exam you took.
 please don't list your 0 grades or Standard Grades here

<u>subject studied</u> (eg typing, computing)	<u>type of exam</u> (eg Pitmans, RSA)	<u>result</u> (eg pass/fail, grade, exam not sat)
.....
.....
.....
.....
.....

1. Did you start a fourth year at school?

tick one box

YES ☐ 1

NO ☐ 2 → if NO, please go to question 6

if YES:

2. Did you spend your third and your fourth years in the same school?

tick one box

yes ☐ 1

no ☐ 2

3. How often did you play truant in your fourth year?

tick one box

never ☐ 1

a lesson here and there ☐ 2

a day here and there ☐ 3

several days at a time ☐ 4

weeks at a time ☐ 5

4. On the whole, do you feel that your fourth year at school was worthwhile?

tick one box

yes ☐ 1

no ☐ 2

5. Did you apply for a Higher School Bursary or grant for staying on into the fifth year?

tick one box

I did not apply for a grant ☐ 1

I applied, but was not offered a grant. ☐ 2

I applied and was offered a grant ☐ 3

I didn't know that I could get a grant. ☐ 4

6. Did you start a fifth year at school?

tick one box

YES ☐ 1

NO ☐ 2 → if NO, please go to page 7

if YES:

7. Why did you start a fifth year?

tick all that apply

I was too young to leave at the end of fourth year. ☐

I wanted qualifications for entering further or higher education. ☐

there were particular courses or subjects I wanted to do. ☐

there were no YTS places available that I wanted. ☐

there were no jobs available that I wanted. ☐

I enjoyed school life ☐

I was too young to enter the job or course I'd chosen ☐

I hadn't decided on my future education or career ☐

I preferred school to the other options open to me at the time. ☐

I had always assumed I would start a fifth year ☐

because my friends were staying on. ☐

I thought that by getting better qualifications I'd improve my job prospects. ☐

I was too young to claim Social Security benefits ☐

8. Which one of these was the most important reason?
please underline it on the list above

1. In your fifth year, did you regularly have classes at another school or college during school hours...
tick one box for each line

	yes	no
...for SCOTVEC modules?	<input type="checkbox"/> 1	<input type="checkbox"/> 2
...for Highers?	<input type="checkbox"/> 1	<input type="checkbox"/> 2
...for other types of exams?	<input type="checkbox"/> 1	<input type="checkbox"/> 2

2. Did you study any subjects for O grade or for Highers in your fifth year at school?
tick one box

if YES: YES ☐ 1

NO ☐ 2 → if NO, please go to the next page

Underline or write in each subject that you studied for O grade or for Highers in fifth year.

tick if you studied the subject for O grade in fifth year
tick if you sat the O grade exam in fifth year and then
ring the exam grade, if any, that you got (n = no award)

do not include any exams you took in your fourth year here

	O grades						Highers								
	tick if studied	tick if sat	grades obtained					tick if studied	tick if sat	grades obtained					
Arithmetic.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Mathematics.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
English.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Chemistry.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Physics.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Biology.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
French.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Geography.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
History.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Modern Studies.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Art and Design.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Secretarial Studies	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Accounting.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Food and Nutrition.	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Technical Drawing..	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
Other subjects (please say which)	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n
.....	<input type="checkbox"/>	<input type="checkbox"/>	1	2	3	4	5	n	<input type="checkbox"/>	<input type="checkbox"/>	A	B	C	D	n

Now, do the same for the subjects you studied for Highers in your fifth year.
(n = no award)

Please check that you have included all the subjects you started for O grade or for Highers, even those you dropped before the exam

1. Altogether, how many SCOTVEC modules did you complete by the end of your fifth year at school?
if none, write '0'

.....modules

2. Did you study for any other type of exam in your fifth year at school, such as CSEs (Certificate of Secondary Education), Standard Grade, GCSEs, SCOTVEC modules or City & Guilds (CGLI)?
tick one box

YES ☐ 1

NO ☐ 2 → if NO, please go to question 3

if YES:

Please write down the name of each subject, the qualification you studied for and the result of any exam you took.
please don't list your O grades or Highers here

<u>subject studied</u> (eg typing, computing)	<u>type of exam</u> (eg Pitmans, RSA)	<u>result</u> (eg pass/fail, grade, exam not sat)
.....
.....
.....
.....
.....

3. Did you receive a Higher School Bursary or grant in your fifth year?
tick one box

yes. ☐ 1
no ☐ 2

if YES: Would you have stayed on into the fifth year if you had not received such a grant?
tick one box

yes. ☐ 1
no ☐ 2

4. On reflection, do you think that staying on into the fifth year was the right decision for you?
tick one box

yes. ☐ 1
no ☐ 2

5. Did your parents want you to complete your fifth year?
tick one box

yes. ☐ 1
no ☐ 2

6. Did any of your teachers advise you to complete your fifth year?
tick one box

yes. ☐ 1
no ☐ 2

1. Did you think about applying for any courses at college or university, before last November (1988)?
include part-time courses
tick one box

YES ☐ 1

NO ☐ 2 → if NO, please go to the next page

if YES:

2. Here is a list of different types of courses, including part-time courses. Please tick any that you...

...considered applying for, before last November (1988)
...applied for, before last November
(including applications later withdrawn)
...were accepted for, before last November
...entered, before last November

	<u>considered</u>	<u>applied for</u>	<u>was accepted</u>	<u>entered</u>
<u>Universities:</u>				
degree courses.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<u>Central Institutions,</u>				
<u>Polytechnics,</u>				
<u>Colleges of Education</u>				
<u>& other colleges:</u>				
degree courses.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
teaching degree courses.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
other advanced courses				
(eg HND, HNC)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SCOTVEC National Certificate				
courses (modules)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
other courses				
(eg Highers, City & Guilds)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<u>Nursing or physiotherapy</u>				
<u>training</u>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<u>Other education or training</u> ..	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. If you applied for any full-time or part-time course for this session, please name the course and institution you most wanted to enter:

Course: please give the main qualification to which the course will lead, and the subject(s)

qualification

subject(s)

Institution: (eg Aberdeen
University, Queen's College)

4. Is this a full-time course?
tick one box

yes. . . . ☐ 1
no ☐ 2

1. Over the next few years, do you hope to...
don't count anything you are already doing
tick one box for each line

	yes	no	not sure
...start a degree course at university, or somewhere else?	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
...start another full-time college course?	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
...start on the Youth Training Scheme (YTS)?	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
...start some other type of training for a job?	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3

2. Have you left school?
tick one box

YES ☐ 1

NO ☐ 2 → if NO, please go to the next page

if YES:

When did you leave school?

month:

year: 19

3. How anxious were your parents that you should do well
in your school work?
tick one box

very anxious.	<input type="checkbox"/> 1
fairly anxious.	<input type="checkbox"/> 2
they were happy as long as I did my best.	<input type="checkbox"/> 3
they didn't mind one way or the other	<input type="checkbox"/> 4

4. Do you think that leaving school then
was the right decision for you?
tick one box

yes.	<input type="checkbox"/> 1
no	<input type="checkbox"/> 2

5. On the whole, do you feel your last year
at school was worthwhile?
tick one box

yes.	<input type="checkbox"/> 1
no	<input type="checkbox"/> 2

6. Why did you leave school when you did?
tick all the reasons that applied

I did not think I could get any better exam results at school.	<input type="checkbox"/>
I needed the money from a job or from the Youth Training Scheme (YTS).	<input type="checkbox"/>
I had all the exam passes needed for the job or course I wanted to do.	<input type="checkbox"/>
I was fed up with school	<input type="checkbox"/>
I wanted to start a particular job while I had the chance.	<input type="checkbox"/>
I wanted to join the Youth Training Scheme (YTS)	<input type="checkbox"/>
I wanted to go to college or university.	<input type="checkbox"/>
my parents advised me to leave school.	<input type="checkbox"/>
I felt I would prefer life at work or college to life at school.	<input type="checkbox"/>
I wanted to get a job and be self-supporting	<input type="checkbox"/>
I couldn't get a bursary or grant to stay on at school	<input type="checkbox"/>

7. Which one of these was the most important reason?
please underline it on the list above

8. Did you have a full-time job fixed up
before you left school?
tick one box

yes.	<input type="checkbox"/> 1
no	<input type="checkbox"/> 2

- Under the Action Plan, many courses in schools, colleges and training centres now include National Certificate modules, sometimes known as 16+ or SCOTVEC modules.

Have you ever started any SCOTVEC National Certificate modules?

tick one box

if YES: YES ☐ 1

NO ☐ 2 → if NO, please go to question 2

By July 1989, how many SCOTVEC National Certificate modules will you have completed altogether...
count all the modules you have ever done
if none, write '0'

number
completed

...while still at school? ☐
...while on a YTS scheme? ☐
...while at college full-time (but not on YTS)? ☐
...while at college part-time
(but not on YTS and not still at school)? ☐

- Over the past year, have you applied for, or started, any courses or training?

count part-time courses

tick one box

if YES: YES ☐ 1

NO ☐ 2 → if NO, please go to the next page

- Over the past year, which types of courses or training have you applied for?

include part-time courses

tick all that apply

	applied for	started
SCOTVEC National Certificate courses (modules) at college	<input type="checkbox"/>	<input type="checkbox"/>
other courses at college (eg Highers, City & Guilds, RSA)	<input type="checkbox"/>	<input type="checkbox"/>
nursing or physiotherapy training at a hospital	<input type="checkbox"/>	<input type="checkbox"/>
training through the Youth Training Scheme (YTS) (but <u>not an apprenticeship</u>).	<input type="checkbox"/>	<input type="checkbox"/>
apprenticeship or technician training as <u>part of YTS</u>	<input type="checkbox"/>	<input type="checkbox"/>
apprenticeship or technician training with a firm (but <u>not on YTS</u>).	<input type="checkbox"/>	<input type="checkbox"/>
apprenticeship or technician training in the Army, Navy or Air Force.	<input type="checkbox"/>	<input type="checkbox"/>
some other education or training.	<input type="checkbox"/>	<input type="checkbox"/>

Now, go back and tick the ones that you have started on
if you haven't started on any, please go to the next page

- Did you get any financial help towards doing your present or most recent course or training?

tick one box on each line

	yes	no	not sure
a training allowance paid by YTS or an employer	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
wages paid by an employer	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
a bursary or grant paid to you by the Region or Council	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
a grant paid to you by an employer or sponsor	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
fees paid to the college by an employer or sponsor.	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
money given to you or fees paid by your parents	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3
some other sort of financial help	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3

1. Have you ever started any course or off-the-job training, perhaps at a college, training centre or university or with your employer?
 count part-time courses
 don't count any courses you took while at school
 tick one box

YES ☐ 1

NO ☐ 2 → if NO, please go to the next page

if YES:

2. Please tell us about your present or most recent course or off-the-job training.
 if you have started a programme of National Certificate modules (sometimes known as 16+ or SCOTVEC modules) tell us about your main area of study

name of the course
or main subjects studied
 eg Catering, Electrical Engineering

qualification(s), if any, studied for
 eg SCOTVEC modules, HNC, BSc.
 if none, write 'none'

name of college or training centre(s)
 eg St Andrew's University, Glasgow School of Art

The next few questions are about your present or most recent course or off-the-job training.

3. When did you start this course? month:
 year: 19.....

4. Is (or was) this course made up of SCOTVEC National Certificate modules?
 tick one box
- | | |
|---------------------|----------------------------|
| all of it | <input type="checkbox"/> 1 |
| some of it. | <input type="checkbox"/> 2 |
| none of it. | <input type="checkbox"/> 3 |
| not sure. | <input type="checkbox"/> 4 |

5. Is (or was) this course part of YTS?
 tick one box
- | | |
|--------------|----------------------------|
| yes. | <input type="checkbox"/> 1 |
| no | <input type="checkbox"/> 2 |

6. Did you get a bursary or grant to do this course?
 tick one box
- | | |
|--------------|----------------------------|
| yes. | <input type="checkbox"/> 1 |
| no | <input type="checkbox"/> 2 |

7. How long does (or did) this course last from beginning to end?
 give the full length of time
 if not known, write 'don't know'
-yrs.....mths.....wks

8. Is (or was) this course...
 tick all that apply

...full-time?	<input type="checkbox"/>	...evening?	<input type="checkbox"/>
...sandwich?	<input type="checkbox"/>	...a 12 or 21 hour course?	<input type="checkbox"/>
...block release?	<input type="checkbox"/>	...other part-time?	<input type="checkbox"/>
...day release?	<input type="checkbox"/>		

9. Are you still taking this course?
 tick one box
- | | |
|--------------|----------------------------|
| yes. | <input type="checkbox"/> 1 |
| no | <input type="checkbox"/> 2 |

1. In the past three months,
how many jobs have you applied for?
if none, write '0'

.....jobs

2. In the past three months,
how have you tried to find a job (or another job)?
tick all that apply

Have you...

...looked at adverts in the paper
or in shop windows?

☐
☐
☐
☐

...gone to the Job Centre?

...approached employers directly?

...asked your parents?

...asked people at work or
on your YTS scheme?

☐
☐
☐
☐

...asked friends or relatives?

...gone to the Careers Office?

...not been looking?

3. What job, if any, do you hope to be doing
in five or six years' time?
if you plan not to be in a job, write 'no job'

.....
.....

4. Have you ever been offered a place
on the Youth Training Scheme (YTS)?
tick one box

yes. ☐ 1
no ☐ 2

5. Have you ever turned down the offer of
a place on the Youth Training Scheme?
tick one box

yes. ☐ 1
no ☐ 2

if YES: Why was this?

.....
.....

6. How many full-time jobs have you had since leaving school?
if you haven't had a full-time job, write '0'
don't count YTS traineeships here

.....jobs

The next few questions are for people who have had a job since
leaving school or have been on the Youth Training Scheme (YTS)
if you haven't had a job and haven't been on YTS,
please go to the next page

7. Have you earned any qualifications, or credits towards a
qualification, while you were doing a job or scheme?
tick one box

YES ☐ 1

NO ☐ 2 → if NO, please go to
the next page

if YES:

Which qualification are (or were) you working towards?
(eg SCOTVEC National Certificate, Caterbase)

name of qualification.....

8. Have you completed all the work for this
qualification yet?
tick one box

yes. ☐ 1
no ☐ 2

1. Are you doing any paid work now, either in a job or on the Youth Training Scheme (YTS)?
count any full-time or part-time work
tick one box

YES ☐ 1

NO ☐ 2 → if NO, please go to the next page

if YES:

2. In what sort of business do you now work?
(eg hospital, shop, college, building firm)

What is the name of your job or YTS scheme?
(eg stores-clerk, nurse)

Describe the actual work you do in your job or scheme
.....
.....

3. Last week, what was your pay from this job or scheme?
before stoppages £.....:.....p
(eg tax and National Insurance) give weekly amount

4. How many hours did you work last week in your job or scheme, not counting lunch breaks?
if none, write '0'hours

5. Are you doing a recognised apprenticeship?
tick one box yes ☐ 1
no ☐ 2
not sure ☐ 3

6. Do you (or did you) get any training or instruction in this job or scheme, either at work or somewhere else?
tick one box

YES ☐ 1

NO ☐ 2 → if NO, please go to the next page

if YES:

7. What sort of training do (or did) you get?
tick yes or no for each one

	yes	no
apprenticeship training.	<input type="checkbox"/> 1	<input type="checkbox"/> 2
courses run by your employer	<input type="checkbox"/> 1	<input type="checkbox"/> 2
courses run at an outside training centre.	<input type="checkbox"/> 1	<input type="checkbox"/> 2
courses run at college	<input type="checkbox"/> 1	<input type="checkbox"/> 2
training from your supervisor or training officer.	<input type="checkbox"/> 1	<input type="checkbox"/> 2
training from workmates.	<input type="checkbox"/> 1	<input type="checkbox"/> 2
other training	<input type="checkbox"/> 1	<input type="checkbox"/> 2

8. How long does (or did) your main training last from beginning to end?
give the full length of timeyrs.....mths.....wks
if not known, write 'don't know'

1. Have you ever started on the Youth Training Scheme (YTS)?
count any scheme you are now on
tick one box

YES ☐ 1

NO ☐ 2 → if NO, please go to
the next page

if YES:

When did you begin your first
YTS scheme?

month.....

year: 19.....

2. Have you left your first scheme?
tick one box

yes. ☐ 1

no ☐ 2

if YES: Why did you leave this scheme?

.....
.....

3. How many YTS schemes
have you ever started on?
count any scheme you are now on
tick one box

one. ☐ 1

two. ☐ 2

three or more. ☐ 3

4. While you were on YTS were you...
tick one box

...an employee?. ☐ 1

...a trainee?. ☐ 2

...not sure? ☐ 3

5. Are you now on a YTS scheme?
tick one box

YES ☐ 1

NO ☐ 2 → if NO, please go to
the next page

if YES:

6. What, if anything, do you like most about your scheme?

.....
.....
.....
.....

7. What, if anything, do you dislike most about your scheme?

.....
.....
.....
.....

8. Since you started this scheme,
have you asked to be moved
to another type of YTS scheme?
tick one box

yes. ☐ 1

no ☐ 2

9. When your scheme ends,
do you think you will be
kept on by your YTS
employer or sponsor?
tick one box

yes ☐ 1

no. ☐ 2

not sure. ☐ 3

1. How many brothers and sisters do you have?
if none, write '0'brotherssisters

2. In your fourth year at secondary school, who did you usually stay with during the school week?
tick one box

mother and father. ☐ 1
mother and step-father ☐ 2
father and step-mother ☐ 3
mother only. ☐ 4

father only. ☐ 5
boarding school ☐ 6
school hostel ☐ 7
other ☐ 8

3. Who do you stay with now?
please underline it on the list above

4. In what type of house do your parents (or step-parents) live?
tick one box

somewhere they own. ☐ 1
somewhere rented from the Council ☐ 2
somewhere rented from the Scottish Special Housing Association (SSHA) ☐ 3
somewhere rented from a housing association other than the SSHA ☐ 4
somewhere rented privately. ☐ 5
some other type of housing. ☐ 6

5. Have you left home?
tick one box

yes. ☐ 1
no ☐ 2

6. How much did you give towards board and keep last week ('dig money')?
if nothing, write '0'

£.....:.....p
give weekly amount

7. Putting together all the money that you get, how much do you get in an average week?
(eg wages, benefits, grants)
if nothing, write '0'

£.....:.....p
give weekly amount

8. Over the past year, have your parents regularly helped you with money?
tick one box

yes. ☐ 1
no ☐ 2

9. Do you get any benefits in your own name?
include the dole and any Social Security benefits
tick one box

yes. ☐ 1
no ☐ 2

10. Are you...
tick one box

...single? ☐ 1
...engaged? ☐ 2
...living with someone? ☐ 3
...married? ☐ 4

11. Do you have any children?
tick one box

yes ☐ 1
no ☐ 2
expecting a baby. ☐ 3

12. Do you have a disability or handicap which would affect the types of work you could do?
tick one box

yes. ☐ 1
no ☐ 2

The next few questions are about your parents, or step-parents

1. Are they now...
tick one box for each

	mother	father
...in full-time paid employment?	<input type="checkbox"/> 1	<input type="checkbox"/> 1
...in part-time paid employment?	<input type="checkbox"/> 2	<input type="checkbox"/> 2
...unemployed and looking for work?	<input type="checkbox"/> 3	<input type="checkbox"/> 3
...retired?	<input type="checkbox"/> 4	<input type="checkbox"/> 4
...full-time working in the home (eg housewife)?	<input type="checkbox"/> 5	<input type="checkbox"/> 5
...unable to work (eg disabled)?	<input type="checkbox"/> 6	<input type="checkbox"/> 6
...doing something else full-time (eg on a course)?	<input type="checkbox"/> 7	<input type="checkbox"/> 7
...dead?	<input type="checkbox"/> 8	<input type="checkbox"/> 8
...don't know.	<input type="checkbox"/> 9	<input type="checkbox"/> 9

2. Please tell us about your parents' (or step-parents') jobs.
if they are not working at the moment,
please tell us about the most recent job each has held

	mother	father
In what type of business do they work? (eg shop, school)
What is the name of the job? (eg shop assistant, teacher) also give any qualification or rank that either parent holds
Please describe briefly what kind of work they do in their jobs

3. Are your mother and father self-employed?
tick one box for each
- | | | |
|---------------|-----------------------------------|-----------------------------------|
| yes | mother <input type="checkbox"/> 1 | father <input type="checkbox"/> 1 |
| no. | mother <input type="checkbox"/> 2 | father <input type="checkbox"/> 2 |

4. How old were your mother and father when they left school?
tick one box for each
- | | | |
|--------------------------------|-----------------------------------|-----------------------------------|
| 15 years old or less | mother <input type="checkbox"/> 1 | father <input type="checkbox"/> 1 |
| 16 years old | mother <input type="checkbox"/> 2 | father <input type="checkbox"/> 2 |
| 17 years old or more | mother <input type="checkbox"/> 3 | father <input type="checkbox"/> 3 |
| don't know | mother <input type="checkbox"/> 4 | father <input type="checkbox"/> 4 |

5. Which qualifications have your mother and father got from courses they took since they left school?
tick all that apply

	mother	father
no qualifications taken since leaving school.	<input type="checkbox"/>	<input type="checkbox"/>
non-advanced qualification (eg ONC, City & Guilds, SEN, Pitmans, TOPS, RSA)	<input type="checkbox"/>	<input type="checkbox"/>
advanced qualification (eg HND, SRN, teaching diploma)	<input type="checkbox"/>	<input type="checkbox"/>
degree or above(eg BSc, BEd, MA)	<input type="checkbox"/>	<input type="checkbox"/>
don't know.	<input type="checkbox"/>	<input type="checkbox"/>

Thank you very much for filling in this questionnaire

We are interested in the Action Plan in Scotland and would like to find out about the modules which young people have taken. To do this, we need to ask SCOTVEC for details of the modules you have done, if any. We guarantee that this information will be treated confidentially and used only for statistical purposes.

If you do not want us to ask SCOTVEC for these details put a cross in the box. If you don't mind please leave it blank

☐

Would you like to tell us more about yourself, in your own words?

What have you been doing over the past year? Have you ever done any SCOTVEC National Certificate modules, at school or college, on YTS, or somewhere else? If so, are you glad you took them? What did you like about the modules you took? What did you dislike about them? Would you want to do more modules in the future?

Even if you have not done any SCOTVEC National Certificate modules, we would like to hear what you think about them.

Please write more on the back of this page

ANNEXE 2 : Significations du schéma Entité/
Association

1. Description des rôles de nos schéma Entité/
Association

Type d'Entité ETUDE :

Associations :

- Liaison : se rapporte à zéro, une ou plusieurs études
- Liaison : est à la base de zéro, une ou plusieurs études
- A : est supervisée par zéro, un ou plusieurs responsables
- Fonde : se fonde sur une ou plusieurs enquêtes

Type d'Entité ENQUETE :

Associations :

- Fonde : sert de fondement au minimum à une étude
- Possède : est supervisée par zéro, un ou plusieurs responsables
- Explique : est annotée par zéro, une ou plusieurs notes
- Concerne : interroge une ou plusieurs populations

Type d'Entité POPULATION :

Associations :

- Concerne : est interrogée par une ou plusieurs enquêtes
- Extrait : comprend un ou plusieurs échantillons

Type d'Entité ECHANTILLON :

Associations :

- Extrait : est extrait d'une ou plusieurs populations

- Se divise : comporte un ou plusieurs sous-échantillons
- Post-stratification : se stratifie en zéro, une ou plusieurs catégories

Type d'Entité SOUS-ECHANTILLON :

Associations :

- Se divise : est une subdivision de un ou plusieurs échantillons
- Reçoit : est sondé par un ou plusieurs questionnaires

Type d'Entité QUESTIONNAIRE :

Associations :

- Reçoit : est envoyé à un ou plusieurs sous-échantillons
- Contient : contient une ou plusieurs questions
- Est sous : est supervisé par un ou plusieurs responsables
- Se divise : est divisé en zéro, une ou plusieurs pages(thèmes)

Type d'Entité QUESTION :

Associations :

- Contient : est contenue dans un ou plusieurs questionnaires
- Comporte : est contenue dans zéro, une ou plusieurs pages et pour chacune de celles-ci, on connaît la position et la version de la question
- Appartient : est d'un et un seul format
- Possède : possède zéro, une ou plusieurs variables
- Filtre : est référencée par zéro, une ou plusieurs variables d'une question-filtre
- Garnit : est garnie par zéro, une ou plusieurs Q-ligne, A-ligne, H-ligne

Type d'Entité VARIABLE :

Associations :

- Possède : est associée à une ou plusieurs questions
- Filtre : renvoie vers zéro, une ou plusieurs questions
- Référence : réfère à zéro, une ou plusieurs Q-ligne, A-ligne, H-ligne
- Dépend : est liée à zéro, une ou plusieurs variables
- Dépend : est référencée par zéro, une ou plusieurs variables
- Possède : possède zéro, une ou plusieurs possibilités de valeurs manquantes
- Définit : est définie par zéro, une ou plusieurs fréquences
- Décrit : est décrite par zéro ou une note

Type d'Entité BASE DE DONNEES :

Associations :

- Appartient : contient une ou plusieurs variables
- Décrit : est décrite par zéro ou une note
- Se situe : se situe dans une ou plusieurs localisations

Type d'Entité RESPONSABLE :

Associations :

- A : est responsable de zéro, une ou plusieurs études
- Possède : est responsable de zéro, une ou plusieurs enquêtes
- Est sous : est responsable de zéro, un ou plusieurs questionnaires

Type d'Entité PAGE/THEME :

Associations :

- Se divise : est une partie de un ou plusieurs questionnaires
- Comporte : comporte au moins une question dont on connaît la version et la position sur la page

Type d'Entité TYPE DE QUESTION :

Associations :

- Associe : caractérise un ou plusieurs formats

Type d'Entité FORMAT :

Associations :

- Appartient : est le format de zéro, une ou plusieurs questions
- Associe : est associé à un et un seul type général de question

Types d'Entités QLIGNE, ALIGNE, HLIGNE :

Associations :

- Garnit : garnit une et une seule question
- Référence : est associée à une ou plusieurs variables

Type d'Entité VALEUR MANQUANTE :

Associations :

- Possède : est possédée par zéro, une ou plusieurs variables

Type d'Entité FREQUENCE :

Associations :

- Définit : définit zéro, une ou plusieurs variables

Type d'Entité CATEGORIE :

Associations :

- Post-stratification : est une strate de zéro, un ou plusieurs échantillons

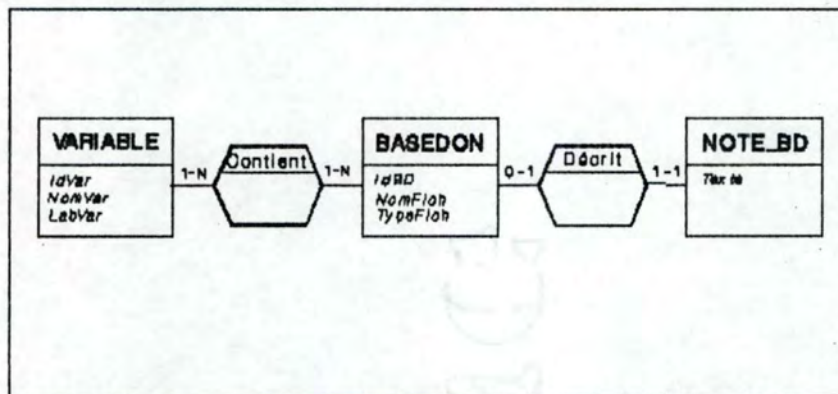
Type d'Entité LOCALISATION :

Associations :

- Se situe : est la machine où se situe zéro, une ou plusieurs bases de données

2. Exemple du schéma Entité/Association vers le schéma relationnel

Soit le schéma Entité/Association suivant :



Exemple de schéma Entité / Association

1) VARIABLE :

- Attributs :
- Identificateur (IdVar)
 - Nom (NomVar)
 - Label (LabVar)

Association : - Contient :
Une VARIABLE est un élément
d'une ou plusieurs (BASEDON)

2) **BASEDON** :

Attributs : - Identificateur (IdBD)
- Nom du fichier (NomFich)
- Type du fichier (TypeFich)

Association : - Contient :
Une BASEDON contient une ou
plusieurs (VARIABLE)

- Décrit :
Une BASEDON est décrite par
zéro ou une (NOTE_ENQ)

3) **NOTE_ENQ** :

Attribut : - Texte de la note (Texte)

Association : - Décrit :
Une NOTE_ENQ décrit une et
une seule (BASEDON)

A partir de ce schéma Entité/Association, nous obten-
drons le schéma relationnel suivant :

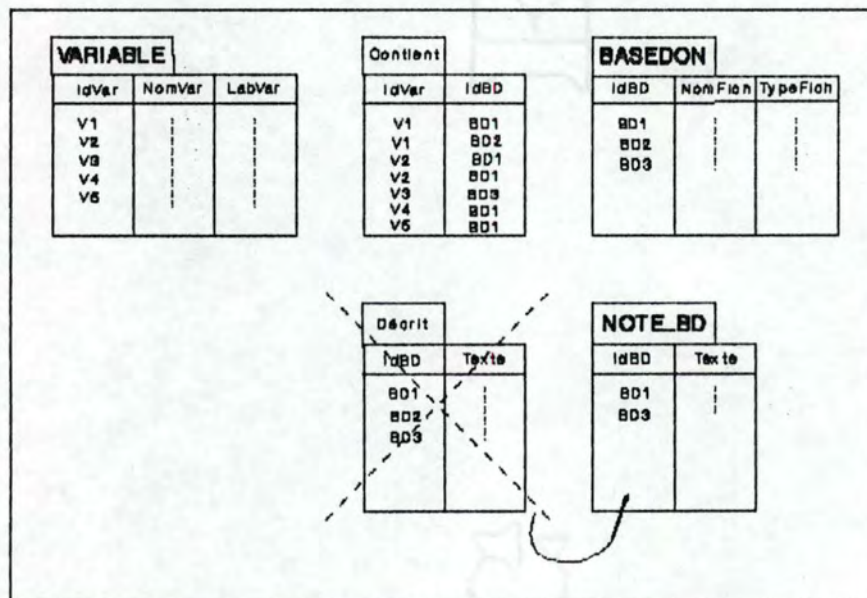


Schéma relationnel correspondant

ANNEXE 3 : Schéma SIR¹ de la base de données DOCPROJ

```

run name          DOCPROJ codebook definition
task name         DOCPROJ codebook initialization commands
old file          DOCPROJ /create /journal=off
password          DOC
task name         DOCPROJ global record type options
max input cols    2192
rectype cols      1,2
max key size      15
max rec types     30
max rec count     1023
n of cases        1
no case id
task name         record 1 (QUESTTXT) schema definition
record schema     1,QUESTTXT /lock
document          That's the question_text table.

```

```

*****
***          TABLE :          QUESTTXT          ***
*****

```

```

sort ids          QMNUMBR      (A)
                  QORDER       (A)
sequence check    off
max rec count     1023
data list         fixed (1) /1  QMNUMBR      3 -      6 (I)
                           /1  QORDER       7 -      8 (I)
                           /1  DBNAME       9 -     16 (A)
                           /1  FORMNUMB    17 -     19 (I)
                           /1  GLOBALNO    20 -     25 (I)
                           /1  WORDING     26 -     28 (I)
                           /1  VARTEXT1    29 -    268 (A)
                           /1  VARTEXT2   269 -    508 (A)
                           /1  VARTEXT3   509 -    748 (A)
                           /1  VARTEXT4   749 -    988 (A)
                           /1  VARTEXT5   989 -   1228 (A)
                           /1  VARTEXT6  1229 -   1468 (A)
                           /1  VARTEXT7  1469 -   1708 (A)
                           /1  VALTEXT1   1709 -   1948 (A)
                           /1  VALTEXT2   1949 -   2188 (A)

var labels        QMNUMBR,      'Number of the question in QM'/
                  QORDER,      'Order of the variable in the
                              question'/
                  DBNAME,      'Name of the variable (from the
                              index)'/
                  FORMNUMB,    'Number of the format of the
                              question'/
                  GLOBALNO,    'Global number of the variable in
                              QM'/
                  WORDING,     'Wording in the question'/
                  VARTEXT1,    'Description of the question
                              asked'/
                  VARTEXT2,    'VARTEXT continued 2'/
                  VARTEXT3,    'VARTEXT continued 3'/
                  VARTEXT4,    'VARTEXT continued 4'/

```

1 L'abréviation SIR est mise pour Scientific Information Retrieval


```

VARTEXT5, 'VARTEXT continued 5'/
VARTEXT6, 'VARTEXT continued 6'/
VARTEXT7, 'VARTEXT continued 7'/
VALTEXT1, 'Descriptions of the potential
           answers'/
VALTEXT2, 'VALTEXT continued'/

end schema
task name      record 2 (ATABLE)  schema definition
record schema  2, ATABLE /lock
document       Containing the A_line (provisional)

```

```

*****
***          TABLE :          ATABLE          ***
*****
sort ids      ANUM          (A)
sequence check off
max rec count 1023
data list     fixed (1) /1  ANUM          3 -      5 (I)
                                   /1  ALINE1          6 -    245 (A)
                                   /1  ALINE2         246 -    485 (A)

var labels    ANUM,         'Number of A_line'/
              ALINE1,      'Line A'/
              ALINE2,      'Line A continued'/

end schema
task name      record 3 (HTABLE)  schema definition
record schema  3, HTABLE /lock
document       Containing the H_line (provisional)

```

```

*****
***          TABLE :          HTABLE          ***
*****
sort ids      HNUM          (A)
sequence check off
max rec count 1023
data list     fixed (1) /1  HNUM          3 -      5 (I)
                                   /1  HLINE1          6 -    245 (A)
                                   /1  HLINE2         246 -    485 (A)

var labels    HNUM,         'Number of the H_line'/
              HLINE1,      'Line H'/
              HLINE2,      'Line H continued'/

end schema
task name      record 4 (QTABLE)  schema definition
record schema  4, QTABLE /lock
document       Containing the Q_line (provisional)

```

```

*****
***          TABLE :          QTABLE          ***
*****
sort ids      QNUM          (A)
sequence check off
max rec count 1023
data list     fixed (1) /1  QNUM          3 -      5 (I)
                                   /1  QLINE1          6 -    245 (A)
                                   /1  QLINE2         246 -    485 (A)
                                   /1  QLINE3         486 -    725 (A)
                                   /1  QLINE4         726 -    965 (A)
                                   /1  QLINE5         966 -   1205 (A)
                                   /1  QLINE6        1206 -   1445 (A)
                                   /1  QLINE7        1446 -   1685 (A)

var labels    QNUM,         'Number of the Q_line'/

```



```

QLINE1, 'Line Q'/
QLINE2, 'Line Q continued'/
QLINE3, 'Line Q continued 1'/
QLINE4, 'Line Q continued 2'/
QLINE5, 'Line Q continued 3'/
QLINE6, 'Line Q continued 4'/
QLINE7, 'Line Q continued 5'/

```

```

end schema
task name      record 5 (QUESVAR)  schema definition
record schema  5,QUESVAR /lock
document       All variables of questionnaires

```

```

*****
***          TABLE :      QUESVAR          ***
*****

```

```

sort ids      IDSUR      (A)
               QMNUMBR   (A)
               QORDER    (A)

sequence check off
max rec count 1023
data list      fixed (1) /1  IDSUR      3 -    10 (A)
                  /1  QMNUMBR   11 -    14 (I)
                  /1  QORDER    15 -    16 (I)
                  /1  DBNAME    17 -    24 (A)
                  /1  FORMNUMB  25 -    27 (I)
                  /1  QMNAME    28 -    44 (A)
                  /1  WORDING   45 -    47 (I)
                  /1  GLOBALNO  48 -    53 (I)

var labels     IDSUR,      'Id of the surveys'/
               QMNUMBR,    'Number of question in QM'/
               QORDER,     'Order of the variable in the
                           question'/
               DBNAME,     'Name of the variable in DB'/
               FORMNUMB,   'Number of the format of the
                           question'/
               QMNAME,     'Name of the variable in QM'/
               WORDING,    'Wording of the question'/
               GLOBALNO,   'Global number of the variable in
                           QM'/

```

```

end schema
task name      record 6 (IND)  schema definition
record schema  6,IND /lock
document       Table containing the INDEX file

```

```

*****
***          TABLE :      IND          ***
*****

```

```

sort ids      QMNUMBR   (A)
               QORDER    (A)

sequence check off
max rec count 2000
data list      fixed (1) /1  QMNUMBR   3 -    6 (I)
                  /1  QORDER    7 -    8 (I)
                  /1  DBNAME    9 -   16 (A)

var labels     QMNUMBR,    'Number of question in QM'/
               QORDER,     'Order of the variable in the
                           question'/
               DBNAME,     'Name of the variable in DB'/

```

```

end schema
task name      record 7 (LISTVAR)  schema definition

```


record schema 7,LISTVAR /lock
document Table containing the QUESVAR file

*** TABLE : LISTVAR ***

sort ids QMNUMBR (A)
QORDER (A)
sequence check off
max rec count 1023
data list fixed (1) /1 QMNUMBR 3 - 6 (I)
/1 QORDER 7 - 8 (I)
/1 DBNAME 9 - 16 (A)
/1 FORMNUMB 17 - 19 (I)
/1 QMNAME 20 - 36 (A)
var labels QMNUMBR, 'Number of the question in QM'/
QORDER, 'Order of the variable in the
question'/
DBNAME, 'Name of the variable in DB'/
FORMNUMB, 'Number of format for the
question'/
QMNAME, 'Name of the variable in QM'/
end schema
task name record 8 (NAMEPROC) schema definition
record schema 8,NAMEPROC /lock
document Containing the procedure names for each
format

*** TABLE : NAMEPROC ***

sort ids CHFORMAT (A)
TYPE (A)
ORDER (A)
sequence check off
max rec count 1023
data list fixed (1) /1 CHFORMAT 3 - 5 (I)
/1 TYPE 6 (A)
/1 ORDER 7 - 10 (I)
/1 PROC1 11 - 18 (A)
/1 PROC2 19 - 26 (A)
/1 PROC3 27 - 34 (A)
/1 PROC4 35 - 42 (A)
/1 PROC5 43 - 50 (A)
/1 PROC6 51 - 58 (A)
/1 PROC7 59 - 66 (A)
/1 PROC8 67 - 74 (A)
/1 PROC9 75 - 82 (A)
/1 PROC10 83 - 90 (A)
/1 PROC11 91 - 98 (A)
/1 PROC12 99 - 106 (A)
/1 PROC13 107 - 114 (A)
/1 PROC14 115 - 122 (A)
/1 PROC15 123 - 130 (A)
/1 PROC16 131 - 138 (A)
/1 PROC17 139 - 146 (A)
/1 PROC18 147 - 154 (A)
/1 PROC19 155 - 162 (A)
/1 PROC20 163 - 170 (A)
var labels CHFORMAT, 'Number of the format of the


```
question'/
TYPE,      'Determines procedures by format
            type'/
ORDER,     'Determines procedures by order of
            vars'/
PROC1,     'Name of procedure 1'/
PROC2,     'Name of procedure 2'/
PROC3,     'Name of procedure 3'/
PROC4,     'Name of procedure 4'/
PROC5,     'Name of procedure 5'/
PROC6,     'Name of procedure 6'/
PROC7,     'Name of procedure 7'/
PROC8,     'Name of procedure 8'/
PROC9,     'Name of procedure 9'/
PROC10,    'Name of procedure 10'/
PROC11,    'Name of procedure 11'/
PROC12,    'Name of procedure 12'/
PROC13,    'Name of procedure 13'/
PROC14,    'Name of procedure 14'/
PROC15,    'Name of procedure 15'/
PROC16,    'Name of procedure 16'/
PROC17,    'Name of procedure 17'/
PROC18,    'Name of procedure 18'/
PROC19,    'Name of procedure 19'/
PROC20,    'Name of procedure 20'/'
```

```
end schema
finish
```


ANNEXE 4 : Spécifications des variables du programme
DOCPROJ

STRLINE : String [*100]
 This variable is GLOBAL
 Represents a line of the Master File obtained in Questmast

STRIND : String [*100]
 This variable is GLOBAL
 Represents a line in the indexed file of the variables (derived from Questmast)

STRQM : String [*100]
 This variable is LOCAL
 Represents the line that will be clean of all Questmast control characters (this is a work variable)

STRCLEAN : String [*100]
 This variable is GLOBAL
 Represents the line STRQM after the cleaning

STRQVAR : String [*100]
 This variable is GLOBAL
 Represents a line of the QUESTVAR file

VARTXT : String [*240] (Array 1..7)
 This variable is GLOBAL
 Represents the text of the asked question

VALTXT : String [*240] (Array 1..2)
 This variable is GLOBAL
 Represents the potential answers to the question

NPROC : String [*8] (Array 1..20)
 This variable is GLOBAL
 Represents the name of the procedures to be executed

SPECHAR : String [*3] (Array 1..9)
 This variable is GLOBAL
 Represents the several control characters that we can find in the Questmast File

LABE : String [*80] (Array 1..130)
 This variable is GLOBAL
 Represents the labels of the errors

ERR : Integer [*2] (Array 1..130)
 This variable is GLOBAL
 Represents the errors (0 if no error, 1 otherwise)

TYPEA : Integer [*2] (Array 1..43)
 This variable is GLOBAL
 Represents the format number of type A

TYPEB : Integer [*2] (Array 1..19)
 This variable is GLOBAL
 Represents the format number of type B

TYPEC : Integer [*2] (Array 1..7)
 This variable is GLOBAL
 Represents the format number of type C

SCHAIN : String [*100]
 This variable is GLOBAL
 Represents a work string

RES1 : String [*100]
 This variable is GLOBAL
 Represents the first set of characters of the initial string

RES2 : String [*100]
 This variable is GLOBAL
 Represents the rest of the initial string

FIRSTVAR : String [*8]
 This variable is GLOBAL
 Represents the first variable in the Questmast File (e.g: in APPUNIVA-APPOTHRDR, APPUNIVA is the first variable)

LASTVAR : String [*8]
 This variable is GLOBAL
 Represents the second variable in the Questmast File (e.g: in APPUNIVA-APPOTHRDR, APPUNIVA is the last variable)

SVY : String [*8]
 This variable is GLOBAL
 Represents a survey

NAMEIND : String [*8]
 This variable is GLOBAL
 Represents the name of the variable in the index file

DBNAMELV : String [*8]
 This variable is GLOBAL
 Represents the name of the variable in the index file

QMNAMELV : String [*17]
 This variable is GLOBAL
 Represents the name of the variable in QuestMast

POSITION : String [*1]
 This variable is GLOBAL
 Represents the first column in the Master File (e.g: Q,A, ,H,V,D,C)

FIRSTPOS : String [*1]
 This variable is GLOBAL
 Represents the first character of a line in the Master File

CHARVAR : String [*1]
This variable is GLOBAL
Represents a work variable

LENGTH : Integer [*4]
This variable is GLOBAL
Represents the length of a string

QUESTNO : Integer [*4]
This variable is GLOBAL
Represents the number of the question

FORMNO : Integer [*4]
This variable is GLOBAL
Represents the format number pf the question

FORMATLV : Integer [*4]
This variable is GLOBAL
Represents the format number of the question

QNIND : Integer [*4]
This variable is GLOBAL
Represents the question number in the index file

QNLV : Integer [*4]
This variable is GLOBAL
Represents the question number in the index file

ORDERIND : Integer [*4]
This variable is GLOBAL
Represents the variable order in the question

ORDERLV : Integer [*4]
This variable is GLOBAL
Represents the variable order in the question

ORD : Integer [*4]
This variable is GLOBAL
Represents the indication of the variables' number

MAXORDER : Integer [*2]
This variable is GLOBAL
Represents the maximum order in a question from
the index

QWVAR : Integer [*4]
This variable is GLOBAL
Represents the number of questions without variable

ACOUNT : Integer [*4]
This variable is GLOBAL
Represents the counter of the A-lines

HCOUNT : Integer [*4]
This variable is GLOBAL
Represents the counter of the H-lines

QCOUNT : Integer [*4]
 This variable is GLOBAL
 Represents the counter of the Q-lines

WORDGCO : Integer [*2]
 This variable is GLOBAL
 Represents the counter of the wordings

GOODPL : Integer [*2]
 This variable is GLOBAL
 Represents the place of the variable in VARTXT

NUMBVT : Integer [*2]
 This variable is GLOBAL
 Represents the maximum number of variables in VARTXT

NUMBVAR : Integer [*2]
 This variable is GLOBAL
 Represents the number of variables in a question
 (e.g: one, two or more)

NUMVARLV : Integer [*2]
 This variable is GLOBAL
 Represents the number of variables in a question
 (e.g: one, two or more)

NUMIND : Integer [*2]
 This variable is GLOBAL
 Represents the number of variables in a question
 (e.g: one, two or more)

NUMFORM : Integer [*2]
 This variable is GLOBAL
 Represents the number of variables required the format

NUMSTR : Integer [*2]
 This variable is GLOBAL
 Represents the number of string to consider

FORMAT : Integer [*4]
 This variable is GLOBAL
 Represents the format number

TYPEFRM : String [*1]
 This variable is GLOBAL
 Represents the documentation type for the format

WRKTXT : String [*240] (Array 1..7)
 This variable is GLOBAL
 Represents the VALTXT (work variable)

STRWRK : String [*100]
 This variable is GLOBAL
 Represents a line (work variable)

WRKNAME : String [*8]
 This variable is GLOBAL
 Represents the variable (work variable)

RES1B : Integer [*2]
This variable is GLOBAL
Represents an indice (work variable)

RES2B : Integer [*2]
This variable is GLOBAL
Represents an indice (work variable)

I : Integer [*4]
This variable is GLOBAL
Represents an indice (work variable)

J : Integer [*4]
This variable is GLOBAL
Represents an indice (work variable)

J1 : Integer [*4]
This variable is GLOBAL
Represents an indice (work variable)

K : Integer [*4]
This variable is GLOBAL
Represents an indice (work variable)

V : Integer [*4]
This variable is GLOBAL
Represents an indice (work variable)

V1 : Integer [*4]
This variable is GLOBAL
Represents an indice (work variable)

FINI : Integer [*2]
This variable is GLOBAL
Represents a boolean which detect the end of a loop
(work variable)

CHECKOK : Integer [*2]
This variable is GLOBAL
Represents a boolean which will take the value 1
if we have to check the index file, 0 otherwise

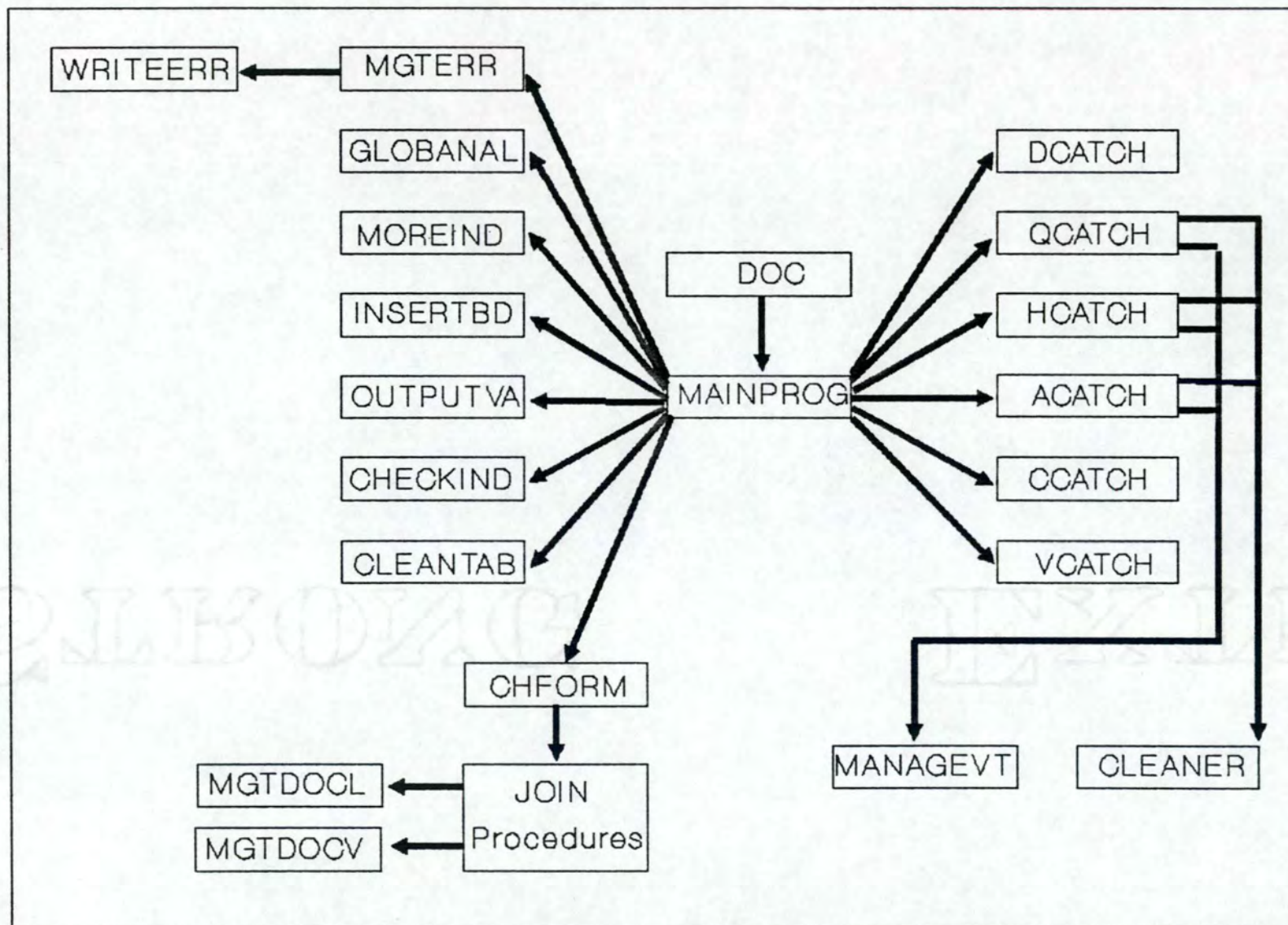


Fig. Annexe 5 : Structure modulaire du programme DOCPROJ


```

C ***** CASE OF A-LINE *****
C
C ACATCH
C
C Ce module a pour objectif de saisir une A-ligne provenant
C du Master File, de la nettoyer des caractères de contrôle
C de QUESTMAST et de la stocker provisoirement dans le
C fichier 'Atable'.
C
C
C
C CALL DBASE.CLEANER
C COMPUTE STRCLEAN = STRQM
C SET VARTXT1 TO VARTXT7 (7 * '')
C IFTHEN (FIRSTPOS = 'A')
C     RECORD IS ATABLE (ACOUNT)
C         IF (LEN(STRCLEAN) GE 2)
C             COMPUTE VARTXT1 =
C                 SBST(STRCLEAN,2,LEN(STRCLEAN)-1)
C             PUT VARS ALINE1 ALINE2 = VARTXT1 VARTXT2
C         END RECORD
C     COMPUTE ACOUNT = ACOUNT+1
C ELSE
C     COMPUTE I = ACOUNT - 1
C     RECORD IS ATABLE (I)
C         GET VARS VARTXT1 VARTXT2 = ALINE1 ALINE2
C         COMPUTE NUMBVT = 2
C         CALL DBASE.MANAGEVT
C         PUT VARS ALINE1 ALINE2 = VARTXT1 VARTXT2
C     END RECORD
C ENDIF
C
C ***** ADDING 'CODER IMPLIED YES' *****
C
C ADDCODER
C
C Ce module a pour objectif d'ajouter dans l'attribut
C VALTEXT de la variable traitée le texte suivant 'coder
C implied yes'.
C
C
C
C COMPUTE WRKTXT1 = 'CODER IMPLIED YES'
C COMPUTE NUMBVT = 2
C COMPUTE NUMSTR = 1
C CALL DBASE.MGTDOCL
C
C ***** ADDING 'NONE TICKED' *****
C
C ADDNOTI
C
C Ce module a pour objectif d'ajouter dans l'attribut
C VALTEXT de la variable traitée le texte suivant 'none
C ticked'.
C

```



```

C
C
. COMPUTE WRKTXT1 = 'NONE TICKED'
. COMPUTE NUMBVT = 2
. COMPUTE NUMSTR = 1
. CALL DBASE.MGTDOCL

C ***** A-LINE FOR VARTEXT *****
C
. COMPUTE J = ORDERLV
. WHILE (J LE (ACOUNT - 1))
.   RECORD IS ATABLE (J)
.     IFTHEN (J NE ORDERLV)
.       CALL DBASE.JOINVW
.     ENDIF
.     GET VARS WRKTXT1 WRKTXT2 = ALINE1 ALINE2
.     COMPUTE NUMBVT = 7
.     COMPUTE NUMSTR = 2
.     CALL DBASE.MGTDOCV
.   END RECORD
.   COMPUTE J = J + NUMVARLV
. END WHILE

C ***** A-LINE AND H-LINE FOR VARTEXT *****
C
. IF (ACOUNT NE 0) J = MOD (ORDERLV,ACOUNT)
. IF (ORDERLV GE ACOUNT) COMPUTE J = J+1
. COMPUTE J1 = 0
. WHILE (J LE (ACOUNT-1))
.   RECORD IS ATABLE (J)
.     IFTHEN (J1 NE 0)
.       CALL DBASE.JOINVW
.     ENDIF
.     COMPUTE J1 = 1
.     GET VARS WRKTXT1 WRKTXT2 = ALINE1 ALINE2
.     COMPUTE NUMBVT = 7
.     COMPUTE NUMSTR = 2
.     CALL DBASE.MGTDOCV
.   END RECORD
.   COMPUTE J = J + ((ACOUNT-1)/WORDGCO)
. END WHILE

C ***** A-LINE FOR VARTEXT *****
C
. COMPUTE J = ORDERLV - 1
. WHILE (J LE (ACOUNT - 1))
.   RECORD IS ATABLE (J)
.     IFTHEN (J NE (ORDERLV - 1))
.       CALL DBASE.JOINVW
.     ENDIF
.     GET VARS WRKTXT1 WRKTXT2 = ALINE1 ALINE2
.     COMPUTE NUMBVT = 7
.     COMPUTE NUMSTR = 2
.     CALL DBASE.MGTDOCV

```



```

.      END RECORD
.      COMPUTE J = J + (NUMVARLV - 1)
.      END WHILE

C      ***** CASE OF C-LINE *****
C
C      NO USE HERE
C

C      ***** CHECK *****
C
C      CHECKIND
C
C      Ce module a pour objectif de vérifier la cohérence des
C      données de l'INDEX vis-à-vis des fichiers Master File et
C      Quesvaret cela chaque fois qu'une question est traitée. Le
C      cas échéant, cette procédure positionne les erreurs le
C      vecteur destiné à cette effet.
C
C      _____
C
C      COMPUTE ORDERIND = 0
C      COMPUTE I = 0
C      V = SRST(QMNAMELV, '-')
C      IF (V = 0) V = SRST(QMNAMELV, ',')
C      IF (V = 0) WRKNAME = SBST(QMNAMELV, 1, 8)
C      IF (V NE 0) WRKNAME = SBST(QMNAMELV, 1, -V-1)
C      COMPUTE WRKNAME = TRIMLR(WRKNAME)
C      IFTHEN (LEN(WRKNAME) NE 0)
C          PROCESS RECORD IND WITH (QNLV)
C              GET VARS ORDERIND NAMEIND = QORDER DBNAME
C              IFTHEN (WRKNAME = NAMEIND)
C                  COMPUTE I = 1
C              ENDIF
C      END RECORD
C      COMPUTE MAXORDER = ORDERIND
C      IF (I = 0)
C          COMPUTE ERR1 TO ERR130 (104) = 1
C      ENDIF
C      IF (QNLV NE QNIND)
C          COMPUTE ERR1 TO ERR130 (105) = 1
C      IF (V = 0) WRKNAME = ''
C      IF (V NE 0) WRKNAME = TRIMLR(SBST(QMNAMELV, -V+1, 8))
C      IFTHEN (LEN(WRKNAME) NE 0)
C          COMPUTE I = 0
C          PROCESS RECORD IND WITH (QNLV)
C              GET VARS ORDERIND NAMEIND = QORDER DBNAME
C              IFTHEN (WRKNAME = NAMEIND)
C                  COMPUTE I = 1
C              ENDIF
C      END RECORD
C      IF (I = 0)
C          COMPUTE ERR1 TO ERR130 (106) = 1
C      IF (QNLV NE QNIND)
C          COMPUTE ERR1 TO ERR130 (105) = 1
C      COMPUTE MAXORDER = ORDERIND

```


ENDIF

***** DETERMINE NUMBER OF VAR *****

CHFORM

Ce module a un objectif triple :

- déterminer le nombre théorique de variables pour une question de tel format
- déterminer la liste des procédures à appliquer successivement pour documenter les variables de la question
- exécuter la liste des procédures nécessaires à la documentation des variables de la question.

```

      COMPUTE NUMFORM = -2
      IFTHEN (FORMATLV = 1)
        COMPUTE NUMFORM = -1
      ELSEIF (FORMATLV = 2)
        COMPUTE NUMFORM = 1
      ELSEIF (FORMATLV = 3)
      ELSEIF (FORMATLV = 4)
        COMPUTE NUMFORM = 1
      ELSEIF (FORMATLV = 5)
      ELSEIF (FORMATLV = 6)
        COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
      ELSEIF (FORMATLV = 7)
      ELSEIF (FORMATLV = 8)
      ELSEIF (FORMATLV = 9)
        COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
      ELSEIF (FORMATLV = 10)
      ELSEIF (FORMATLV = 11)
      ELSEIF (FORMATLV = 12)
        COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
      ELSEIF (FORMATLV = 13)
      ELSEIF (FORMATLV = 14)
      ELSEIF (FORMATLV = 15)
      ELSEIF (FORMATLV = 16)
      ELSEIF (FORMATLV = 17)
      ELSEIF (FORMATLV = 18)
      ELSEIF (FORMATLV = 19)
        COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
      ELSEIF (FORMATLV = 20)
      ELSEIF (FORMATLV = 21)
      ELSEIF (FORMATLV = 22)
        COMPUTE NUMFORM = 1
      ELSEIF (FORMATLV = 23)
        COMPUTE NUMFORM = 2
      ELSEIF (FORMATLV = 24)
      ELSEIF (FORMATLV = 25)
      ELSEIF (FORMATLV = 26)
      ELSEIF (FORMATLV = 27)
      ELSEIF (FORMATLV = 28)
        COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
      ELSEIF (FORMATLV = 29)
        COMPUTE NUMFORM = 1
      ELSEIF (FORMATLV = 30)

```



```

. ELSEIF (FORMATLV = 31)
. ELSEIF (FORMATLV = 32)
. ELSEIF (FORMATLV = 33)
.     COMPUTE NUMFORM = 2
. ELSEIF (FORMATLV = 34)
.     COMPUTE NUMFORM = 2
. ELSEIF (FORMATLV = 35)
.     COMPUTE NUMFORM = 1
. ELSEIF (FORMATLV = 36)
.     COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 37)
.     COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 38)
. ELSEIF (FORMATLV = 39)
. ELSEIF (FORMATLV = 40)
. ELSEIF (FORMATLV = 41)
. ELSEIF (FORMATLV = 42)
. ELSEIF (FORMATLV = 43)
.     COMPUTE NUMFORM = ((ACOUNT - 1)/WORDGCO)*
.                         ((HCOUNT - 1)/WORDGCO)
.
. ELSEIF (FORMATLV = 44)
. ELSEIF (FORMATLV = 45)
. ELSEIF (FORMATLV = 46)
.     COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 47)
. ELSEIF (FORMATLV = 48)
. ELSEIF (FORMATLV = 49)
.     COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 50)
.     COMPUTE NUMFORM = 1
. ELSEIF (FORMATLV = 51)
. ELSEIF (FORMATLV = 52)
. ELSEIF (FORMATLV = 53)
. ELSEIF (FORMATLV = 54)
. ELSEIF (FORMATLV = 55)
. ELSEIF (FORMATLV = 56)
. ELSEIF (FORMATLV = 57)
.     COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 58)
. ELSEIF (FORMATLV = 59)
. ELSEIF (FORMATLV = 60)
.     COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 61)
.     COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 62)
.     COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 63)
.     COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 64)
. ELSEIF (FORMATLV = 65)
.     COMPUTE NUMFORM = ((ACOUNT - 1)/WORDGCO) + 1
. ELSEIF (FORMATLV = 66)
.     COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 67)
. ELSEIF (FORMATLV = 68)
. ELSEIF (FORMATLV = 69)
.     COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 70)
.     COMPUTE NUMFORM = 1
. ELSEIF (FORMATLV = 71)

```



```

.      COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 72)
.      COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 73)
. ELSEIF (FORMATLV = 74)
. ELSEIF (FORMATLV = 75)
.      COMPUTE NUMFORM = 1
. ELSEIF (FORMATLV = 76)
.      COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 77)
.      COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 78)
.      COMPUTE NUMFORM = 1
. ELSEIF (FORMATLV = 79)
.      COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 80)
.      COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 81)
.      COMPUTE NUMFORM = 1
. ELSEIF (FORMATLV = 82)
. ELSEIF (FORMATLV = 83)
. ELSEIF (FORMATLV = 84)
. ELSEIF (FORMATLV = 85)
.      COMPUTE NUMFORM = 1
. ELSEIF (FORMATLV = 86)
. ELSEIF (FORMATLV = 87)
. ELSEIF (FORMATLV = 88)
. ELSEIF (FORMATLV = 89)
. ELSEIF (FORMATLV = 90)
.      COMPUTE NUMFORM = ((ACOUNT - 1)/WORDGCO) + 1
. ELSEIF (FORMATLV = 91)
. ELSEIF (FORMATLV = 92)
.      COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 93)
.      COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
. ELSEIF (FORMATLV = 94)
. ELSEIF (FORMATLV = 95)
. ELSEIF (FORMATLV = 96)
.      COMPUTE NUMFORM = ((ACOUNT - 1)/WORDGCO) + 1
. ELSEIF (FORMATLV = 97)
.      COMPUTE NUMFORM = ((ACOUNT - 1)/WORDGCO) + 1
. ELSEIF (FORMATLV = 98)
.      COMPUTE NUMFORM = 6
. ELSEIF (FORMATLV = 99)
. ELSEIF (FORMATLV = 100)
.      COMPUTE NUMFORM = ((ACOUNT - 1)/WORDGCO) + 1
. ELSEIF (FORMATLV = 101)
. ELSEIF (FORMATLV = 102)
. ELSEIF (FORMATLV = 103)
.      COMPUTE NUMFORM = ((ACOUNT - 1)/WORDGCO) + 1
. ELSEIF (FORMATLV = 104)
.      COMPUTE NUMFORM = ((ACOUNT - 1)/WORDGCO)*
.                          ((HCOUNT - 1)/WORDGCO)
. ELSEIF (FORMATLV = 105)
.      COMPUTE NUMFORM = -1
. ELSEIF (FORMATLV = 106)
.      COMPUTE NUMFORM = -1
. ELSEIF (FORMATLV = 107)
.      COMPUTE NUMFORM = ((ACOUNT - 1)/WORDGCO)*
.                          ((HCOUNT - 1)/WORDGCO)

```



```

.      ELSEIF (FORMATLV = 108)
.          COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
.      ELSEIF (FORMATLV = 109)
.          COMPUTE NUMFORM = 1
.      ELSEIF (FORMATLV = 110)
.          COMPUTE NUMFORM = 1
.      ELSEIF (FORMATLV = 111)
.      ELSEIF (FORMATLV = 112)
.          COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
.      ELSEIF (FORMATLV = 113)
.          COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
.      ELSEIF (FORMATLV = 114)
.          COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
.      ELSEIF (FORMATLV = 115)
.          COMPUTE NUMFORM = (ACOUNT - 1)/WORDGCO
.      ELSEIF (FORMATLV = 116)
.          COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
.      ELSEIF (FORMATLV = 117)
.      ELSEIF (FORMATLV = 118)
.          COMPUTE NUMFORM = 3
.      ELSEIF (FORMATLV = 119)
.          COMPUTE NUMFORM = 3
.      ELSEIF (FORMATLV = 120)
.          COMPUTE NUMFORM = 3
.      ELSEIF (FORMATLV = 121)
.          COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
.      ELSEIF (FORMATLV = 122)
.          COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
.      ELSEIF (FORMATLV = 123)
.          COMPUTE NUMFORM = ((ACOUNT - 1)/WORDGCO) + 1
.      ELSEIF (FORMATLV = 124)
.          COMPUTE NUMFORM = 1
.      ELSEIF (FORMATLV = 125)
.      ELSEIF (FORMATLV = 126)
.          COMPUTE NUMFORM = (HCOUNT - 1)/WORDGCO
.      ELSEIF (FORMATLV = 127)
.          COMPUTE NUMFORM = ((ACOUNT - 1)/WORDGCO) + 1
.      ELSEIF (FORMATLV = 128)
.          COMPUTE NUMFORM = -1
.      ENDIF
C -----
.      SET VARTXT1 TO VARTXT7 (7 * '')
.      SET VALTXT1 TO VALTXT2 (2 * '')
.      COMPUTE V = 0
.      IFTHEN (SRCH(TYPEA1,TYPEA43,FORMATLV) GT 0)
.          COMPUTE CHARVAR = 'A'
.          RECORD IS NAMEPROC (FORMATLV,CHARVAR,V)
.              GET VARS NPROC1 NPROC2 NPROC3 NPROC4 NPROC5 NPROC6
.                  NPROC7 NPROC8 NPROC9 NPROC10 NPROC11
.                  NPROC12 NPROC13 NPROC14 NPROC15 NPROC16
.                  NPROC17 NPROC18 NPROC19 NPROC20 = PROC1
.                  PROC2 PROC3 PROC4 PROC5 PROC6 PROC7 PROC8
.                  PROC9 PROC10 PROC11 PROC12 PROC13 PROC14
.                  PROC15 PROC16 PROC17 PROC18 PROC19 PROC20
.      END RECORD
.      ELSE
.          IFTHEN (SRCH(TYPEB1,TYPEB19,FORMATLV) GT 0)
.              COMPUTE CHARVAR = 'B'
.              IFTHEN (NUMVARLV = 1)
.                  COMPUTE V = 1

```



```

.      RECORD IS NAMEPROC (FORMATLV,CHARVAR,V)
.      GET VARS NPROC1 NPROC2 NPROC3 NPROC4 NPROC5
.              NPROC6 NPROC7 NPROC8 NPROC9 NPROC10
.              NPROC11 NPROC12 NPROC13 NPROC14
.              NPROC15 NPROC16 NPROC17 NPROC18
.              NPROC19 NPROC20 = PROC1 PROC2 PROC3
.              PROC4 PROC5 PROC6 PROC7 PROC8 PROC9
.              PROC10 PROC11 PROC12 PROC13 PROC14
.              PROC15 PROC16 PROC17 PROC18 PROC19
.              PROC20
.
.      END RECORD
.
.      ELSE
.      COMPUTE V = 3
.      RECORD IS NAMEPROC (FORMATLV,CHARVAR,V)
.      GET VARS NPROC1 NPROC2 NPROC3 NPROC4 NPROC5
.              NPROC6 NPROC7 NPROC8 NPROC9 NPROC10
.              NPROC11 NPROC12 NPROC13 NPROC14
.              NPROC15 NPROC16 NPROC17 NPROC18
.              NPROC19 NPROC20 = PROC1 PROC2 PROC3
.              PROC4 PROC5 PROC6 PROC7 PROC8 PROC9
.              PROC10 PROC11 PROC12 PROC13 PROC14
.              PROC15 PROC16 PROC17 PROC18 PROC19
.              PROC20
.
.      END RECORD
.
.      ENDIF
.
.      ELSE
.      IFTHEN (SRCH(TYPEC1,TYPEC7,FORMATLV) GT 0)
.      COMPUTE CHARVAR = 'C'
.      IFTHEN (ORDERLV = 1)
.      RECORD IS NAMEPROC (FORMATLV,CHARVAR,ORDERLV)
.      GET VARS NPROC1 NPROC2 NPROC3 NPROC4 NPROC5
.              NPROC6 NPROC7 NPROC8 NPROC9 NPROC10
.              NPROC11 NPROC12 NPROC13 NPROC14
.              NPROC15 NPROC16 NPROC17 NPROC18
.              NPROC19 NPROC20 = PROC1 PROC2 PROC3
.              PROC4 PROC5 PROC6 PROC7 PROC8 PROC9
.              PROC10 PROC11 PROC12 PROC13 PROC14
.              PROC15 PROC16 PROC17 PROC18 PROC19
.              PROC20
.
.      END RECORD
.
.      ELSE
.      COMPUTE V = 3
.      RECORD IS NAMEPROC (FORMATLV,CHARVAR,V)
.      GET VARS NPROC1 NPROC2 NPROC3 NPROC4 NPROC5
.              NPROC6 NPROC7 NPROC8 NPROC9 NPROC10
.              NPROC11 NPROC12 NPROC13 NPROC14
.              NPROC15 NPROC16 NPROC17 NPROC18
.              NPROC19 NPROC20 = PROC1 PROC2 PROC3
.              PROC4 PROC5 PROC6 PROC7 PROC8 PROC9
.              PROC10 PROC11 PROC12 PROC13 PROC14
.              PROC15 PROC16 PROC17 PROC18 PROC19
.              PROC20
.
.      END RECORD
.
.      ENDIF
.
.      ENDIF
.
.      ENDIF
.      COMPUTE K = 1
.      WHILE ((K LE 19) AND (LEN(NPROC1 TO NPROC20(K)) NE 0))
.      IFTHEN (NPROC1 TO NPROC20(K) = 'ADDCODER')

```



```

.      CALL DBASE.ADDCODER
C. WRITE 'ADDCODER'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'ADDNOTI')
.      CALL DBASE.ADDNOTI
C. WRITE 'ADDNOTI'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'ALINEV')
.      CALL DBASE.ALINEV
C. WRITE 'ALINEV'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'ALINEVH')
.      CALL DBASE.ALINEVH
C. WRITE 'ALINEVH'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'ALINEVO')
.      CALL DBASE.ALINEVO
C. WRITE 'ALINEVO'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'DEPENDON')
.      CALL DBASE.DEPENDON
C. WRITE 'DEPENDON'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'HLINEV')
.      CALL DBASE.HLINEV
C. WRITE 'HLINEV'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'HLINEVA')
.      CALL DBASE.HLINEVA
C. WRITE 'HLINEVA'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'HLINEVO')
.      CALL DBASE.HLINEVO
C. WRITE 'HLINEVO'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'JOINLT')
.      CALL DBASE.JOINLT
C. WRITE 'JOINLT'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'JOINLX')
.      CALL DBASE.JOINLX
C. WRITE 'JOINLX'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'JOINVT')
.      CALL DBASE.JOINVT
C. WRITE 'JOINVT'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'JOINVX')
.      CALL DBASE.JOINVX
C. WRITE 'JOINVX'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'QLINEV')
.      CALL DBASE.QLINEV
C. WRITE 'QLINEV'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'SALINEL')
.      CALL DBASE.SALINEL
C. WRITE 'SALINEL'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'SALINEV')
.      CALL DBASE.SALINEV
C. WRITE 'SALINEV'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'SHLINEL')
.      CALL DBASE.SHLINEL
C. WRITE 'SHLINEL'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'SHLINEV')
.      CALL DBASE.SHLINEV
C. WRITE 'SHLINEV'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'SQLINEV')
.      CALL DBASE.SQLINEV
C. WRITE 'SQLINEV'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'TICKED')
.      CALL DBASE.TICKED
C. WRITE 'TICKED'
.      ELSEIF (NPROC1 TO NPROC20(K) = 'YESNO')

```



```

.      CALL DBASE.YESNO
.      ENDIF
.      COMPUTE K = K + 1
.      END WHILE

C      ***** CLEAN A LINE *****
C
C      CLEANER
C
C      Ce module a pour objectif de nettoyer une ligne de texte
C      provenant du Master File des caractères de contrôle de
C      Questmast.
C
C      _____
C
C      COMPUTE STRQM = TRIMLR(PACK(STRLINE))
C      LENGTH = LEN(STRQM)
C      COMPUTE I=1
C      WHILE (I LE 9)
C          COMPUTE V=1
C          WHILE (V NE 0)
C              V = SRST(STRQM,SPECHAR1 TO SPECHAR9(I))
C              V=ABS(V)
C              IFTHEN ((I GE 7) OR (V EQ 0))
C                  IFTHEN ((V NE 0) AND (I = 9))
C                      IFTHEN ((V+1) NE LENGTH)
C                          COMPUTE STRQM = SBST(STRQM,1,V-1) +
C                          '$'+SBST(STRQM,V+2,LENGTH-V-1)
C                      ELSE
C                          COMPUTE STRQM =
C                          SBST(STRQM,1,LENGTH-2)
C                      ENDIF
C                  ENDIF
C              IFTHEN ((V NE 0) AND ((I = 8) OR (I = 7)))
C                  IFTHEN ((V+2) NE LENGTH)
C                      COMPUTE STRQM = SBST(STRQM,1,V-1) +
C                      '$'+SBST(STRQM,V+3,LENGTH-V-2)
C                  ELSE
C                      COMPUTE STRQM =
C                      SBST(STRQM,1,LENGTH-3)
C                  ENDIF
C              ENDIF
C          ELSE
C              IFTHEN (I = 1)
C                  COMPUTE STRQM = SBST(STRQM,1,V-1) +
C                  SBST(STRQM,V+2,LENGTH-V-1)
C              ELSE
C                  IFTHEN (V NE LENGTH)
C                      IFTHEN ((V - 1) NE 0)
C                          COMPUTE STRQM = SBST(STRQM,1,V-1) +
C                          SBST(STRQM,V+1,LENGTH-V)
C                      ELSE
C                          COMPUTE STRQM =
C                          SBST(STRQM,2,LENGTH-1)
C                      ENDIF
C                  ELSE
C                      COMPUTE STRQM =
C                      SBST(STRQM,1,LENGTH-1)

```



```

.               ENDIF
.           ENDIF
.       ENDIF
.       COMPUTE LENGTH = LEN(STRQM)
.   END WHILE
.   COMPUTE I = I+1
. END WHILE

```

```

C ***** CLEAN THE TABLE *****
C
C
C

```

CLEANTAB

```

C Ce module a pour objectif de réinitialiser toutes les
C variables de travail nécessaires au traitement d'une
C question ainsi que d'initialiser à vide les fichiers de
C stockage provisoire 'Qtable', 'Atable', 'Htable'.
C
C
C

```

```

. COMPUTE WRKTXT1 = ''
. COMPUTE I = 1
. WHILE (I LT ACOUNT)
.     RECORD IS ATABLE (I)
.         PUT VARS ALINE1 ALINE2 = WRKTXT1 WRKTXT1
.     END RECORD
.     COMPUTE I = I + 1
. END WHILE
. COMPUTE I = 1
. WHILE (I LT HCOUNT)
.     RECORD IS HTABLE (I)
.         PUT VARS HLINE1 HLINE2 = WRKTXT1 WRKTXT1
.     END RECORD
.     COMPUTE I = I + 1
. END WHILE
. COMPUTE I = 1
. WHILE (I LT QCOUNT)
.     RECORD IS QTABLE (I)
.         PUT VARS QLINE1 TO QLINE7 = WRKTXT1 WRKTXT1
.         WRKTXT1 WRKTXT1 WRKTXT1 WRKTXT1
.         WRKTXT1
.     END RECORD
.     COMPUTE I = I + 1
. END WHILE
. COMPUTE HCOUNT = 1
. COMPUTE ACOUNT = 1
. COMPUTE QCOUNT = 1
. COMPUTE WORDGCO = 1
. COMPUTE NUMVARLV = 0
. SET ERR1 TO ERR130 (130 * 0)

```



```

C ***** ALL CASES OF FORMAT *****
C
C CPROC
C
C Ce module a pour objectif de créer un fichier 'Nameproc'
C contenant la liste des procédures à exécuter pour docu-
C menter chaque question quel que soit son format.
C
C
C RETRIEVAL UPDATE
C . INTEGER *4 ORD,FORMAT
C . STRING *8 NPROC1 TO NPROC20
C . STRING *1 TYPEFRM
C -----
C . COMPUTE FORMAT = 1
C . WHILE (FORMAT LE 128)
C .     SET NPROC1 TO NPROC20 (20 * '')
C .     COMPUTE TYPEFRM = 'A'
C .     COMPUTE ORD = 0
C .     IFTHEN (FORMAT = 1)
C .         COMPUTE NPROC1 = 'SQLINEV'
C .         COMPUTE NPROC2 = 'DEPENDON'
C .     ELSEIF (FORMAT = 2)
C .         COMPUTE NPROC1 = 'SQLINEV'
C .         COMPUTE NPROC2 = 'SALINEL'
C .     ELSEIF (FORMAT = 3)
C .     ELSEIF (FORMAT = 4)
C .         COMPUTE NPROC1 = 'SQLINEV'
C .         COMPUTE NPROC2 = 'SHLINEL'
C .         COMPUTE NPROC3 = 'JOINLT'
C .         COMPUTE NPROC4 = 'ADDNOTI'
C .     ELSEIF (FORMAT = 5)
C .     ELSEIF (FORMAT = 6)
C .         COMPUTE NPROC1 = 'ALINEV'
C .         COMPUTE NPROC2 = 'SHLINEL'
C .     ELSEIF (FORMAT = 7)
C .     ELSEIF (FORMAT = 8)
C .     ELSEIF (FORMAT = 9)
C .         COMPUTE TYPEFRM = 'B'
C .         COMPUTE ORD = 1
C .         COMPUTE NPROC1 = 'SQLINEV'
C .         COMPUTE NPROC2 = 'JOINVT'
C .         COMPUTE NPROC3 = 'SALINEV'
C .         COMPUTE NPROC4 = 'DEPENDON'
C .         NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
C .             PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
C .                 PROC6 PROC7 PROC8 PROC9 PROC10
C .                 PROC11 PROC12 PROC13 PROC14 PROC15
C .                 PROC16 PROC17 PROC18 PROC19 PROC20
C .                 = NPROC1 NPROC2 NPROC3 NPROC4
C .                 NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
C .                 NPROC10 NPROC11 NPROC12 NPROC13
C .                 NPROC14 NPROC15 NPROC16 NPROC17
C .                 NPROC18 NPROC19 NPROC20
C .
C .         END RECORD
C .         COMPUTE ORD = 3
C .         COMPUTE NPROC1 = 'SQLINEV'
C .         COMPUTE NPROC2 = 'JOINVT'

```



```

.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.
.      ELSEIF (FORMAT = 10)
.      ELSEIF (FORMAT = 11)
.      ELSEIF (FORMAT = 12)
.
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.
.      ELSEIF (FORMAT = 13)
.      ELSEIF (FORMAT = 14)
.      ELSEIF (FORMAT = 15)
.      ELSEIF (FORMAT = 16)
.      ELSEIF (FORMAT = 17)
.      ELSEIF (FORMAT = 18)
.      ELSEIF (FORMAT = 19)
.
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'HLINEV'
.      COMPUTE NPROC4 = 'SALINEL'
.
.      ELSEIF (FORMAT = 20)
.      ELSEIF (FORMAT = 21)
.      ELSEIF (FORMAT = 22)
.
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'SHLINEL'
.      COMPUTE NPROC3 = 'JOINLT'
.      COMPUTE NPROC4 = 'ADDCODER'
.
.      ELSEIF (FORMAT = 23)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.
.      ELSEIF (FORMAT = 24)
.      ELSEIF (FORMAT = 25)
.      ELSEIF (FORMAT = 26)
.      ELSEIF (FORMAT = 27)
.      ELSEIF (FORMAT = 28)
.
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.
.      ELSEIF (FORMAT = 29)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'SHLINEL'
.
.      ELSEIF (FORMAT = 30)
.      ELSEIF (FORMAT = 31)
.      ELSEIF (FORMAT = 32)
.      ELSEIF (FORMAT = 33)
.
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.
.      ELSEIF (FORMAT = 34)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.
.      ELSEIF (FORMAT = 35)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'SALINEL'
.
.      ELSEIF (FORMAT = 36)
.      COMPUTE TYPEFRM = 'B'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'

```



```

.      COMPUTE NPROC4 = 'DEPENDON'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.              PROC6 PROC7 PROC8 PROC9 PROC10
.              PROC11 PROC12 PROC13 PROC14 PROC15
.              PROC16 PROC17 PROC18 PROC19 PROC20
.              = NPROC1 NPROC2 NPROC3 NPROC4
.              NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.              NPROC10 NPROC11 NPROC12 NPROC13
.              NPROC14 NPROC15 NPROC16 NPROC17
.              NPROC18 NPROC19 NPROC20
.
.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.      ELSEIF (FORMAT = 37)
.      COMPUTE TYPEFRM = 'B'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.              PROC6 PROC7 PROC8 PROC9 PROC10
.              PROC11 PROC12 PROC13 PROC14 PROC15
.              PROC16 PROC17 PROC18 PROC19 PROC20
.              = NPROC1 NPROC2 NPROC3 NPROC4
.              NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.              NPROC10 NPROC11 NPROC12 NPROC13
.              NPROC14 NPROC15 NPROC16 NPROC17
.              NPROC18 NPROC19 NPROC20
.
.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'HLINEV'
.      COMPUTE NPROC4 = 'JOINVT'
.      COMPUTE NPROC5 = 'ALINEV'
.      COMPUTE NPROC6 = 'DEPENDON'
.      ELSEIF (FORMAT = 38)
.      ELSEIF (FORMAT = 39)
.      ELSEIF (FORMAT = 40)
.      ELSEIF (FORMAT = 41)
.      ELSEIF (FORMAT = 42)
.      ELSEIF (FORMAT = 43)
.      COMPUTE TYPEFRM = 'B'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SHLINEV'
.      COMPUTE NPROC4 = 'JOINVT'
.      COMPUTE NPROC5 = 'SALINEV'
.      COMPUTE NPROC6 = 'TICKED'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.              PROC6 PROC7 PROC8 PROC9 PROC10
.              PROC11 PROC12 PROC13 PROC14 PROC15
.              PROC16 PROC17 PROC18 PROC19 PROC20
.              = NPROC1 NPROC2 NPROC3 NPROC4

```



```

NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
NPROC10 NPROC11 NPROC12 NPROC13
NPROC14 NPROC15 NPROC16 NPROC17
NPROC18 NPROC19 NPROC20

END RECORD
COMPUTE ORD = 3
COMPUTE NPROC1 = 'SQLINEV'
COMPUTE NPROC2 = 'JOINVT'
COMPUTE NPROC3 = 'HLINEV'
COMPUTE NPROC4 = 'JOINVT'
COMPUTE NPROC5 = 'ALINEV'
COMPUTE NPROC6 = 'TICKED'
ELSEIF (FORMAT = 44)
ELSEIF (FORMAT = 45)
ELSEIF (FORMAT = 46)
COMPUTE TYPEFRM = 'B'
COMPUTE ORD = 1
COMPUTE NPROC1 = 'SQLINEV'
COMPUTE NPROC2 = 'JOINVT'
COMPUTE NPROC3 = 'SHLINEV'
COMPUTE NPROC4 = 'SALINEL'
NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
PROC6 PROC7 PROC8 PROC9 PROC10
PROC11 PROC12 PROC13 PROC14 PROC15
PROC16 PROC17 PROC18 PROC19 PROC20
= NPROC1 NPROC2 NPROC3 NPROC4
NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
NPROC10 NPROC11 NPROC12 NPROC13
NPROC14 NPROC15 NPROC16 NPROC17
NPROC18 NPROC19 NPROC20

END RECORD
COMPUTE ORD = 3
COMPUTE NPROC1 = 'SQLINEV'
COMPUTE NPROC2 = 'JOINVT'
COMPUTE NPROC3 = 'HLINEV'
COMPUTE NPROC4 = 'SALINEL'
ELSEIF (FORMAT = 47)
ELSEIF (FORMAT = 48)
ELSEIF (FORMAT = 49)
COMPUTE TYPEFRM = 'B'
COMPUTE ORD = 1
COMPUTE NPROC1 = 'SQLINEV'
COMPUTE NPROC2 = 'JOINVT'
COMPUTE NPROC3 = 'SALINEV'
COMPUTE NPROC4 = 'DEPENDON'
NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
PROC6 PROC7 PROC8 PROC9 PROC10
PROC11 PROC12 PROC13 PROC14 PROC15
PROC16 PROC17 PROC18 PROC19 PROC20
= NPROC1 NPROC2 NPROC3 NPROC4
NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
NPROC10 NPROC11 NPROC12 NPROC13
NPROC14 NPROC15 NPROC16 NPROC17
NPROC18 NPROC19 NPROC20

END RECORD
COMPUTE ORD = 3
COMPUTE NPROC1 = 'SQLINEV'
COMPUTE NPROC2 = 'JOINVT'

```



```

.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.  ELSEIF (FORMAT = 50)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'SALINEL'
.  ELSEIF (FORMAT = 51)
.  ELSEIF (FORMAT = 52)
.  ELSEIF (FORMAT = 53)
.  ELSEIF (FORMAT = 54)
.  ELSEIF (FORMAT = 55)
.  ELSEIF (FORMAT = 56)
.  ELSEIF (FORMAT = 57)
.      COMPUTE TYPEFRM = 'C'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.  NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.          PROC6 PROC7 PROC8 PROC9 PROC10
.          PROC11 PROC12 PROC13 PROC14 PROC15
.          PROC16 PROC17 PROC18 PROC19 PROC20
.          = NPROC1 NPROC2 NPROC3 NPROC4
.          NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.          NPROC10 NPROC11 NPROC12 NPROC13
.          NPROC14 NPROC15 NPROC16 NPROC17
.          NPROC18 NPROC19 NPROC20
.
.  END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'HLINEVO'
.      COMPUTE NPROC4 = 'DEPENDON'
.  ELSEIF (FORMAT = 58)
.  ELSEIF (FORMAT = 59)
.  ELSEIF (FORMAT = 60)
.      COMPUTE TYPEFRM = 'B'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'SHLINEL'
.  NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.          PROC6 PROC7 PROC8 PROC9 PROC10
.          PROC11 PROC12 PROC13 PROC14 PROC15
.          PROC16 PROC17 PROC18 PROC19 PROC20
.          = NPROC1 NPROC2 NPROC3 NPROC4
.          NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.          NPROC10 NPROC11 NPROC12 NPROC13
.          NPROC14 NPROC15 NPROC16 NPROC17
.          NPROC18 NPROC19 NPROC20
.
.  END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'SHLINEL'
.  ELSEIF (FORMAT = 61)
.      COMPUTE NPROC1 = 'ALINEV'
.      COMPUTE NPROC2 = 'SHLINEL'

```



```

      COMPUTE NPROC3 = 'JOINLT'
      COMPUTE NPROC4 = 'ADDNOTI'
ELSEIF (FORMAT = 62)
      COMPUTE NPROC1 = 'ALINEV'
      COMPUTE NPROC2 = 'SHLINEL'
      COMPUTE NPROC3 = 'JOINLT'
      COMPUTE NPROC4 = 'ADDNOTI'
ELSEIF (FORMAT = 63)
      COMPUTE TYPEFRM = 'B'
      COMPUTE ORD = 1
      COMPUTE NPROC1 = 'SQLINEV'
      COMPUTE NPROC2 = 'JOINVT'
      COMPUTE NPROC3 = 'SHLINEV'
      COMPUTE NPROC4 = 'SALINEL'
      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
              PROC6 PROC7 PROC8 PROC9 PROC10
              PROC11 PROC12 PROC13 PROC14 PROC15
              PROC16 PROC17 PROC18 PROC19 PROC20
              = NPROC1 NPROC2 NPROC3 NPROC4
              NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
              NPROC10 NPROC11 NPROC12 NPROC13
              NPROC14 NPROC15 NPROC16 NPROC17
              NPROC18 NPROC19 NPROC20

      END RECORD
      COMPUTE ORD = 3
      COMPUTE NPROC1 = 'SQLINEV'
      COMPUTE NPROC2 = 'JOINVT'
      COMPUTE NPROC3 = 'HLINEV'
      COMPUTE NPROC4 = 'SALINEL'
ELSEIF (FORMAT = 64)
ELSEIF (FORMAT = 65)
      COMPUTE TYPEFRM = 'C'
      COMPUTE ORD = 1
      COMPUTE NPROC1 = 'SQLINEV'
      COMPUTE NPROC2 = 'SALINEL'
      COMPUTE NPROC3 = 'JOINLT'
      COMPUTE NPROC4 = 'DEPENDON'
      COMPUTE NPROC5 = 'JOINLX'
      COMPUTE NPROC6 = 'TICKED'
      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
              PROC6 PROC7 PROC8 PROC9 PROC10
              PROC11 PROC12 PROC13 PROC14 PROC15
              PROC16 PROC17 PROC18 PROC19 PROC20
              = NPROC1 NPROC2 NPROC3 NPROC4
              NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
              NPROC10 NPROC11 NPROC12 NPROC13
              NPROC14 NPROC15 NPROC16 NPROC17
              NPROC18 NPROC19 NPROC20

      END RECORD
      COMPUTE ORD = 3
      COMPUTE NPROC1 = 'SQLINEV'
      COMPUTE NPROC2 = 'JOINVT'
      COMPUTE NPROC3 = 'HLINEVO'
      COMPUTE NPROC4 = 'SALINEL'
      COMPUTE NPROC5 = 'JOINLT'
      COMPUTE NPROC6 = 'DEPENDON'
      COMPUTE NPROC7 = 'JOINLX'
      COMPUTE NPROC8 = 'TICKED'

```



```

. ELSEIF (FORMAT = 66)
.   COMPUTE TYPEFRM = 'B'
.   COMPUTE ORD = 1
.   COMPUTE NPROC1 = 'SQLINEV'
.   COMPUTE NPROC2 = 'JOINVT'
.   COMPUTE NPROC3 = 'SALINEV'
.   COMPUTE NPROC4 = 'DEPENDON'
.   NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.     PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.       PROC6 PROC7 PROC8 PROC9 PROC10
.       PROC11 PROC12 PROC13 PROC14 PROC15
.       PROC16 PROC17 PROC18 PROC19 PROC20
.       = NPROC1 NPROC2 NPROC3 NPROC4
.       NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.       NPROC10 NPROC11 NPROC12 NPROC13
.       NPROC14 NPROC15 NPROC16 NPROC17
.       NPROC18 NPROC19 NPROC20
.
.   END RECORD
.   COMPUTE ORD = 3
.   COMPUTE NPROC1 = 'SQLINEV'
.   COMPUTE NPROC2 = 'JOINVT'
.   COMPUTE NPROC3 = 'ALINEV'
.   COMPUTE NPROC4 = 'DEPENDON'
. ELSEIF (FORMAT = 67)
. ELSEIF (FORMAT = 68)
. ELSEIF (FORMAT = 69)
.   COMPUTE TYPEFRM = 'B'
.   COMPUTE ORD = 1
.   COMPUTE NPROC1 = 'SQLINEV'
.   COMPUTE NPROC2 = 'JOINVT'
.   COMPUTE NPROC3 = 'SALINEV'
.   COMPUTE NPROC4 = 'DEPENDON'
.   NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.     PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.       PROC6 PROC7 PROC8 PROC9 PROC10
.       PROC11 PROC12 PROC13 PROC14 PROC15
.       PROC16 PROC17 PROC18 PROC19 PROC20
.       = NPROC1 NPROC2 NPROC3 NPROC4
.       NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.       NPROC10 NPROC11 NPROC12 NPROC13
.       NPROC14 NPROC15 NPROC16 NPROC17
.       NPROC18 NPROC19 NPROC20
.
.   END RECORD
.   COMPUTE ORD = 3
.   COMPUTE NPROC1 = 'SQLINEV'
.   COMPUTE NPROC2 = 'JOINVT'
.   COMPUTE NPROC3 = 'ALINEV'
.   COMPUTE NPROC4 = 'DEPENDON'
. ELSEIF (FORMAT = 70)
.   COMPUTE NPROC1 = 'SQLINEV'
.   COMPUTE NPROC2 = 'DEPENDON'
. ELSEIF (FORMAT = 71)
.   COMPUTE TYPEFRM = 'B'
.   COMPUTE ORD = 1
.   COMPUTE NPROC1 = 'SQLINEV'
.   COMPUTE NPROC2 = 'JOINVT'
.   COMPUTE NPROC3 = 'SALINEV'
.   COMPUTE NPROC4 = 'SHLINEL'
.   NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.     PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5

```



```

PROC6 PROC7 PROC8 PROC9 PROC10
PROC11 PROC12 PROC13 PROC14 PROC15
PROC16 PROC17 PROC18 PROC19 PROC20
= NPROC1 NPROC2 NPROC3 NPROC4
NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
NPROC10 NPROC11 NPROC12 NPROC13
NPROC14 NPROC15 NPROC16 NPROC17
NPROC18 NPROC19 NPROC20

.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'SHLINEL'
.      ELSEIF (FORMAT = 72)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'HLINEVA'
.      COMPUTE NPROC4 = 'JOINVT'
.      COMPUTE NPROC5 = 'ALINEVH'
.      COMPUTE NPROC6 = 'DEPENDON'
.      ELSEIF (FORMAT = 73)
.      ELSEIF (FORMAT = 74)
.      ELSEIF (FORMAT = 75)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'SALINEL'
.      COMPUTE NPROC3 = 'JOINLT'
.      COMPUTE NPROC4 = 'SHLINEL'
.      ELSEIF (FORMAT = 76)
.      COMPUTE TYPEFRM = 'B'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'SHLINEL'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.      PROC6 PROC7 PROC8 PROC9 PROC10
.      PROC11 PROC12 PROC13 PROC14 PROC15
.      PROC16 PROC17 PROC18 PROC19 PROC20
.      = NPROC1 NPROC2 NPROC3 NPROC4
.      NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.      NPROC10 NPROC11 NPROC12 NPROC13
.      NPROC14 NPROC15 NPROC16 NPROC17
.      NPROC18 NPROC19 NPROC20

.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'SHLINEL'
.      ELSEIF (FORMAT = 77)
.      COMPUTE TYPEFRM = 'B'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'SHLINEL'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5

```



```

PROC6 PROC7 PROC8 PROC9 PROC10
PROC11 PROC12 PROC13 PROC14 PROC15
PROC16 PROC17 PROC18 PROC19 PROC20
= NPROC1 NPROC2 NPROC3 NPROC4
NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
NPROC10 NPROC11 NPROC12 NPROC13
NPROC14 NPROC15 NPROC16 NPROC17
NPROC18 NPROC19 NPROC20

.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'SHLINEL'
.      ELSEIF (FORMAT = 78)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'SALINEL'
.      ELSEIF (FORMAT = 79)
.      COMPUTE TYPEFRM = 'B'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'TICKED'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.      PROC6 PROC7 PROC8 PROC9 PROC10
.      PROC11 PROC12 PROC13 PROC14 PROC15
.      PROC16 PROC17 PROC18 PROC19 PROC20
.      = NPROC1 NPROC2 NPROC3 NPROC4
.      NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.      NPROC10 NPROC11 NPROC12 NPROC13
.      NPROC14 NPROC15 NPROC16 NPROC17
.      NPROC18 NPROC19 NPROC20

.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'TICKED'
.      ELSEIF (FORMAT = 80)
.      COMPUTE TYPEFRM = 'B'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'TICKED'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.      PROC6 PROC7 PROC8 PROC9 PROC10
.      PROC11 PROC12 PROC13 PROC14 PROC15
.      PROC16 PROC17 PROC18 PROC19 PROC20
.      = NPROC1 NPROC2 NPROC3 NPROC4
.      NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.      NPROC10 NPROC11 NPROC12 NPROC13
.      NPROC14 NPROC15 NPROC16 NPROC17
.      NPROC18 NPROC19 NPROC20

.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'

```



```

.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'TICKED'
.  ELSEIF (FORMAT = 81)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'SALINEL'
.  ELSEIF (FORMAT = 82)
.  ELSEIF (FORMAT = 83)
.  ELSEIF (FORMAT = 84)
.  ELSEIF (FORMAT = 85)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'YESNO'
.  ELSEIF (FORMAT = 86)
.  ELSEIF (FORMAT = 87)
.  ELSEIF (FORMAT = 88)
.  ELSEIF (FORMAT = 89)
.  ELSEIF (FORMAT = 90)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'TICKED'
.  ELSEIF (FORMAT = 91)
.  ELSEIF (FORMAT = 92)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'HLINEVA'
.      COMPUTE NPROC4 = 'JOINVT'
.      COMPUTE NPROC5 = 'ALINEVH'
.      COMPUTE NPROC6 = 'DEPENDON'
.  ELSEIF (FORMAT = 93)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'HLINEVA'
.      COMPUTE NPROC4 = 'JOINVT'
.      COMPUTE NPROC5 = 'ALINEVH'
.      COMPUTE NPROC6 = 'DEPENDON'
.  ELSEIF (FORMAT = 94)
.  ELSEIF (FORMAT = 95)
.  ELSEIF (FORMAT = 96)
.      COMPUTE TYPEFRM = 'C'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.      COMPUTE NPROC3 = 'JOINLX'
.      COMPUTE NPROC4 = 'TICKED'
.  NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.          PROC6 PROC7 PROC8 PROC9 PROC10
.          PROC11 PROC12 PROC13 PROC14 PROC15
.          PROC16 PROC17 PROC18 PROC19 PROC20
.          = NPROC1 NPROC2 NPROC3 NPROC4
.          NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.          NPROC10 NPROC11 NPROC12 NPROC13
.          NPROC14 NPROC15 NPROC16 NPROC17
.          NPROC18 NPROC19 NPROC20
.
.  END RECORD
.  COMPUTE ORD = 3
.  COMPUTE NPROC1 = 'SQLINEV'
.  COMPUTE NPROC2 = 'JOINVT'
.  COMPUTE NPROC3 = 'ALINEVO'

```



```

.      COMPUTE NPROC4 = 'DEPENDON'
.      COMPUTE NPROC5 = 'JOINLX'
.      COMPUTE NPROC6 = 'TICKED'
.      ELSEIF (FORMAT = 97)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.      COMPUTE NPROC5 = 'JOINLX'
.      COMPUTE NPROC6 = 'TICKED'
.      ELSEIF (FORMAT = 98)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.      COMPUTE NPROC5 = 'JOINLX'
.      COMPUTE NPROC6 = 'TICKED'
.      ELSEIF (FORMAT = 99)
.      ELSEIF (FORMAT = 100)
.      COMPUTE TYPEFRM = 'C'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.      COMPUTE NPROC3 = 'JOINLX'
.      COMPUTE NPROC4 = 'TICKED'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.      PROC6 PROC7 PROC8 PROC9 PROC10
.      PROC11 PROC12 PROC13 PROC14 PROC15
.      PROC16 PROC17 PROC18 PROC19 PROC20
.      = NPROC1 NPROC2 NPROC3 NPROC4
.      NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.      NPROC10 NPROC11 NPROC12 NPROC13
.      NPROC14 NPROC15 NPROC16 NPROC17
.      NPROC18 NPROC19 NPROC20
.
.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEVO'
.      COMPUTE NPROC4 = 'DEPENDON'
.      COMPUTE NPROC5 = 'JOINLX'
.      COMPUTE NPROC6 = 'TICKED'
.      ELSEIF (FORMAT = 101)
.      ELSEIF (FORMAT = 102)
.      ELSEIF (FORMAT = 103)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SHLINEV'
.      COMPUTE NPROC4 = 'JOINVT'
.      COMPUTE NPROC5 = 'SALINEV'
.      COMPUTE NPROC6 = 'DEPENDON'
.      ELSEIF (FORMAT = 104)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'HLINEVA'
.      COMPUTE NPROC4 = 'JOINVT'
.      COMPUTE NPROC5 = 'ALINEVH'
.      COMPUTE NPROC6 = 'DEPENDON'
.      COMPUTE NPROC7 = 'JOINLX'

```



```

.      COMPUTE NPROC8 = 'TICKED'
.  ELSEIF (FORMAT = 105)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.  ELSEIF (FORMAT = 106)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.  ELSEIF (FORMAT = 107)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'SHLINEL'
.      COMPUTE NPROC5 = 'JOINLT'
.      COMPUTE NPROC6 = 'DEPENDON'
.      COMPUTE NPROC7 = 'JOINLX'
.      COMPUTE NPROC8 = 'TICKED'
.  ELSEIF (FORMAT = 108)
.      COMPUTE TYPEFRM = 'B'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.      COMPUTE NPROC5 = 'JOINLX'
.      COMPUTE NPROC6 = 'TICKED'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.          PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.                  PROC6 PROC7 PROC8 PROC9 PROC10
.                  PROC11 PROC12 PROC13 PROC14 PROC15
.                  PROC16 PROC17 PROC18 PROC19 PROC20
.                  = NPROC1 NPROC2 NPROC3 NPROC4
.                  NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.                  NPROC10 NPROC11 NPROC12 NPROC13
.                  NPROC14 NPROC15 NPROC16 NPROC17
.                  NPROC18 NPROC19 NPROC20
.
.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'ALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.      COMPUTE NPROC5 = 'JOINLX'
.      COMPUTE NPROC6 = 'TICKED'
.  ELSEIF (FORMAT = 109)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'SALINEL'
.  ELSEIF (FORMAT = 110)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'SHLINEL'
.      COMPUTE NPROC3 = 'JOINLT'
.      COMPUTE NPROC4 = 'ADDCODER'
.  ELSEIF (FORMAT = 111)
.  ELSEIF (FORMAT = 112)
.      COMPUTE TYPEFRM = 'B'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'SHLINEL'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)

```



```

      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
      PROC6 PROC7 PROC8 PROC9 PROC10
      PROC11 PROC12 PROC13 PROC14 PROC15
      PROC16 PROC17 PROC18 PROC19 PROC20
      = NPROC1 NPROC2 NPROC3 NPROC4
      NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
      NPROC10 NPROC11 NPROC12 NPROC13
      NPROC14 NPROC15 NPROC16 NPROC17
      NPROC18 NPROC19 NPROC20

      END RECORD
      COMPUTE ORD = 3
      COMPUTE NPROC1 = 'SQLINEV'
      COMPUTE NPROC2 = 'JOINVT'
      COMPUTE NPROC3 = 'ALINEV'
      COMPUTE NPROC4 = 'SHLINEL'
ELSEIF (FORMAT = 113)
      COMPUTE TYPEFRM = 'B'
      COMPUTE ORD = 1
      COMPUTE NPROC1 = 'SQLINEV'
      COMPUTE NPROC2 = 'JOINVT'
      COMPUTE NPROC3 = 'SHLINEV'
      COMPUTE NPROC4 = 'SALINEL'
      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
      PROC6 PROC7 PROC8 PROC9 PROC10
      PROC11 PROC12 PROC13 PROC14 PROC15
      PROC16 PROC17 PROC18 PROC19 PROC20
      = NPROC1 NPROC2 NPROC3 NPROC4
      NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
      NPROC10 NPROC11 NPROC12 NPROC13
      NPROC14 NPROC15 NPROC16 NPROC17
      NPROC18 NPROC19 NPROC20

      END RECORD
      COMPUTE ORD = 3
      COMPUTE NPROC1 = 'SQLINEV'
      COMPUTE NPROC2 = 'JOINVT'
      COMPUTE NPROC3 = 'HLINEV'
      COMPUTE NPROC4 = 'SALINEL'
ELSEIF (FORMAT = 114)
      COMPUTE NPROC1 = 'SQLINEV'
      COMPUTE NPROC2 = 'JOINVT'
      COMPUTE NPROC3 = 'ALINEV'
      COMPUTE NPROC4 = 'DEPENDON'
      COMPUTE NPROC5 = 'JOINLX'
      COMPUTE NPROC6 = 'TICKED'
ELSEIF (FORMAT = 115)
      COMPUTE TYPEFRM = 'B'
      COMPUTE ORD = 1
      COMPUTE NPROC1 = 'SALINEV'
      COMPUTE NPROC2 = 'SHLINEL'
      COMPUTE NPROC3 = 'JOINLT'
      COMPUTE NPROC4 = 'ADDNOTI'
      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
      PROC6 PROC7 PROC8 PROC9 PROC10
      PROC11 PROC12 PROC13 PROC14 PROC15
      PROC16 PROC17 PROC18 PROC19 PROC20
      = NPROC1 NPROC2 NPROC3 NPROC4
      NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
      NPROC10 NPROC11 NPROC12 NPROC13

```



```

                                NPROC14 NPROC15 NPROC16 NPROC17
                                NPROC18 NPROC19 NPROC20
.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'ALINEV'
.      COMPUTE NPROC2 = 'SHLINEL'
.      COMPUTE NPROC3 = 'JOINLT'
.      COMPUTE NPROC4 = 'ADDNOTI'
.      ELSEIF (FORMAT = 116)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'HLINEV'
.      COMPUTE NPROC4 = 'SALINEL'
.      ELSEIF (FORMAT = 117)
.      ELSEIF (FORMAT = 118)
.      COMPUTE TYPEFRM = 'C'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.      COMPUTE NPROC3 = 'JOINLX'
.      COMPUTE NPROC4 = 'TICKED'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.      PROC6 PROC7 PROC8 PROC9 PROC10
.      PROC11 PROC12 PROC13 PROC14 PROC15
.      PROC16 PROC17 PROC18 PROC19 PROC20
.      = NPROC1 NPROC2 NPROC3 NPROC4
.      NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.      NPROC10 NPROC11 NPROC12 NPROC13
.      NPROC14 NPROC15 NPROC16 NPROC17
.      NPROC18 NPROC19 NPROC20
.
.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.      COMPUTE NPROC5 = 'JOINLX'
.      COMPUTE NPROC6 = 'TICKED'
.      ELSEIF (FORMAT = 119)
.      COMPUTE TYPEFRM = 'C'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.      COMPUTE NPROC3 = 'JOINLX'
.      COMPUTE NPROC4 = 'TICKED'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.      PROC6 PROC7 PROC8 PROC9 PROC10
.      PROC11 PROC12 PROC13 PROC14 PROC15
.      PROC16 PROC17 PROC18 PROC19 PROC20
.      = NPROC1 NPROC2 NPROC3 NPROC4
.      NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.      NPROC10 NPROC11 NPROC12 NPROC13
.      NPROC14 NPROC15 NPROC16 NPROC17
.      NPROC18 NPROC19 NPROC20
.
.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'

```



```

.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.      COMPUTE NPROC5 = 'JOINLX'
.      COMPUTE NPROC6 = 'TICKED'
.  ELSEIF (FORMAT = 120)
.      COMPUTE TYPEFRM = 'C'
.      COMPUTE ORD = 1
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.      COMPUTE NPROC3 = 'JOINLX'
.      COMPUTE NPROC4 = 'TICKED'
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.      PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.          PROC6 PROC7 PROC8 PROC9 PROC10
.          PROC11 PROC12 PROC13 PROC14 PROC15
.          PROC16 PROC17 PROC18 PROC19 PROC20
.          = NPROC1 NPROC2 NPROC3 NPROC4
.          NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.          NPROC10 NPROC11 NPROC12 NPROC13
.          NPROC14 NPROC15 NPROC16 NPROC17
.          NPROC18 NPROC19 NPROC20
.
.      END RECORD
.      COMPUTE ORD = 3
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.      COMPUTE NPROC5 = 'JOINLX'
.      COMPUTE NPROC6 = 'TICKED'
.  ELSEIF (FORMAT = 121)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.  ELSEIF (FORMAT = 122)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SHLINEV'
.      COMPUTE NPROC4 = 'SALINEL'
.  ELSEIF (FORMAT = 123)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.      COMPUTE NPROC5 = 'JOINLX'
.      COMPUTE NPROC6 = 'TICKED'
.  ELSEIF (FORMAT = 124)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'DEPENDON'
.  ELSEIF (FORMAT = 125)
.  ELSEIF (FORMAT = 126)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SHLINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.  ELSEIF (FORMAT = 127)
.      COMPUTE NPROC1 = 'SQLINEV'
.      COMPUTE NPROC2 = 'JOINVT'
.      COMPUTE NPROC3 = 'SALINEV'
.      COMPUTE NPROC4 = 'DEPENDON'
.      COMPUTE NPROC5 = 'JOINLX'
.      COMPUTE NPROC6 = 'TICKED'

```



```

.      ELSEIF (FORMAT = 128)
.          COMPUTE NPROC1 = 'SQLINEV'
.          COMPUTE NPROC2 = 'JOINVT'
.          COMPUTE NPROC3 = 'SALINEV'
.          COMPUTE NPROC4 = 'DEPENDON'
.      ENDIF
.      NEW RECORD IS NAMEPROC (FORMAT,TYPEFRM,ORD)
.          PUT VARS PROC1 PROC2 PROC3 PROC4 PROC5
.                  PROC6 PROC7 PROC8 PROC9 PROC10
.                  PROC11 PROC12 PROC13 PROC14 PROC15
.                  PROC16 PROC17 PROC18 PROC19 PROC20
.                  = NPROC1 NPROC2 NPROC3 NPROC4
.                  NPROC5 NPROC6 NPROC7 NPROC8 NPROC9
.                  NPROC10 NPROC11 NPROC12 NPROC13
.                  NPROC14 NPROC15 NPROC16 NPROC17
.                  NPROC18 NPROC19 NPROC20
.
.      END RECORD
.      COMPUTE FORMAT = FORMAT + 1
.      END WHILE
WRITE ' ----- NAMEPROC TABLE IS CREATED -----'
END RETRIEVAL

```



```
C ***** CASE OF D-LINE *****
```

```
C DCATCH
```

```
C Ce module a pour objectif de saisir le numéro et le format
```

```
C de la question provenant du Master File.
```

```
C _____
```

```
C . COMPUTE QUESTNO = NUMBR(TRIMLR(SBST(STRLINE,2,3)))
```

```
C . COMPUTE FORMNO = NUMBR(TRIMLR(SBST(STRLINE,11,3)))
```

```
C
```

```
C ***** ADDING 'DEPENDENT ON TOPIC' *****
```

```
C DEPENDON
```

```
C Ce module a pour objectif d'ajouter dans l'attribut
```

```
C VALTEXT de la variable traitée le texte suivant 'dependent
```

```
C on topic of the question'.
```

```
C _____
```

```
C . COMPUTE WRKTXT1 = 'DEPENDENT ON TOPIC OF THE QUESTION'
```

```
C . COMPUTE NUMBVT = 2
```

```
C . COMPUTE NUMSTR = 1
```

```
C . CALL DBASE.MGTDOCL
```

```
C
```

```
C ***** PRG TO RUN *****
```

```
C DOC
```

```
C Ce module a pour objectif de lancer l'exécution de
```

```
C l'ensemble des programmes nécessaires au projet DOCPROJ.
```

```
C _____
```

```
C SET SCREEN NOPAUSE
```

```
C SET REMARK OFF
```

```
C SET DEMAND SURVEY 'Identification of the survey ( <8 CHAR.)'
```

```
C SET DEMAND CHECKOK 'Check INDEX (0 if NO, 1 if YES)'
```

```
C SET DEMAND FMASTER 'Name of the MASTER file '
```

```
C SET DEMAND FQUESVAR 'Name of QUESVAR file containing the DB
```

```
C variables '
```

```
C SET DEMAND FINDEX 'Name of the INDEX file (<return> if any) '
```

```
C RUN DBASE.CLINETAB
```

```
C RUN DBASE.LOADIND
```

```
C RUN DBASE.LOADVAR
```

```
C RUN DBASE.CPROC
```

```
C * WORKING .....
```

```
C RUN DBASE.MAINPROG
```

```
C ESCAPE SET PROTECTION=(G:RWE) DOC_ERR.
```

```
C ESCAPE SET PROTECTION=(G:RWE) DOC_LIST.
```

```
C * SET DEMAND FFICH 'Name of the database saved as sequential
```

```
C * file ?'
```

```
C * X UNLOAD FILE FILENAME=<FFICH>
```



```

C   figurant dans l'Index et pas dans Quesvar.
C
C
C
C   _____
C
C   WRITE ('DOC_ERR') ' '
C   WRITE ('DOC_ERR') 'NUMBER OF QUESTION WITHOUT VARIABLES IS
C   ' QWVAR
C
C   CIF EQ <CHECKOK>,1
C       WRITE ('DOC_ERR') ' '
C       WRITE ('DOC_ERR') 'NUMBER OF QUESTION ONLY IN THE INDEX'
C       WRITE ('DOC_ERR') 'AND NOT IN QUESVAR'
C       WRITE ('DOC_ERR') ' '
C       COMPUTE V = 0
C       PROCESS RECORD IND
C           MOVE VARS QNIND = QMNUMBER
C           COMPUTE I = 0
C           PROCESS RECORD LISTVAR WITH (QNIND)
C               COMPUTE I = 1
C           END RECORD
C           IFTHEN ((I = 0) AND (QNIND GT V))
C               WRITE ('DOC_ERR') QNIND
C           ENDIF
C           IF (QNIND GT V) COMPUTE V = QNIND
C       END RECORD
C   CIF END

```

```

C   ***** CASE OF H-LINE *****
C
C   HCATCH
C
C   Ce module a pour objectif de saisir une H-ligne provenant
C   du Master File, de la nettoyer des caractères de contrôle
C   de QUESTMAST et de la stocker provisoirement dans le
C   fichier 'Htable'.
C
C   _____
C
C   CALL DBASE.CLEANER
C   COMPUTE STRCLEAN = STRQM
C   SET VARTXT1 TO VARTXT7 (7 * ' ')
C   IFTHEN (FIRSTPOS = 'H')
C       RECORD IS HTABLE (HCOUNT)
C           IF (LEN(STRCLEAN) GE 2)
C               COMPUTE VARTXT1 =
C               SBST(STRCLEAN,2,LEN(STRCLEAN)-1)
C           PUT VARS HLINE1 HLINE2 = VARTXT1 VARTXT2
C       END RECORD
C       COMPUTE HCOUNT = HCOUNT+1
C   ELSE
C       COMPUTE I = HCOUNT - 1
C       RECORD IS HTABLE (I)
C           GET VARS VARTXT1 VARTXT2 = HLINE1 HLINE2
C           COMPUTE NUMBVT = 2
C           CALL DBASE.MANAGEVT
C           PUT VARS HLINE1 HLINE2 = VARTXT1 VARTXT2
C       END RECORD
C   ENDIF

```



```

C ***** H-LINE AND A-LINE FOR VARTEXT *****
C
. J = TRUNC (ORDERLV/ACOUNT) + 1
. COMPUTE J1 = 0
. WHILE (J LE (HCOUNT-1))
.     RECORD IS HTABLE (J)
.         IFTHEN (J1 NE 0)
.             CALL DBASE.JOINVW
.         ENDIF
.         COMPUTE J1 = 1
.         GET VARS WRKTXT1 WRKTXT2 = HLINE1 HLINE2
.         COMPUTE NUMBVT = 7
.         COMPUTE NUMSTR = 2
.         CALL DBASE.MGTDOCV
.     END RECORD
.     COMPUTE J = J + ((HCOUNT-1)/WORDGCO)
. END WHILE

C ***** H-LINE FOR VARTEXT *****
C
. COMPUTE J = ORDERLV - 1
. WHILE (J LE (HCOUNT - 1))
.     RECORD IS HTABLE (J)
.         IFTHEN (J NE (ORDERLV - 1))
.             CALL DBASE.JOINVW
.         ENDIF
.         GET VARS WRKTXT1 WRKTXT2 = HLINE1 HLINE2
.         COMPUTE NUMBVT = 7
.         COMPUTE NUMSTR = 2
.         CALL DBASE.MGTDOCV
.     END RECORD
.     COMPUTE J = J + (NUMVARLV - 1)
. END WHILE

C ***** INSERT INTO THE DB *****
C
C INSERTBD
C
C Ce module a pour objectif de stocker la documentation
C (VARTEXT et VALTEXT) d'une variable dans la base de
C données définitive DOCPROJ uniquement si aucune erreur
C n'est signalée pour cette question. Malgré tout, un
C 'warning' (erreur légère) autorise le sauvetage dans la
C base de données.
C
C
C
C
. COMPUTE I = 0
. COMPUTE V = 100
. WHILE ((V LE 130) AND (I NE 1))
.     COMPUTE V1 = ERR1 TO ERR130 (V)
.     IF (V1 = 1)
.         COMPUTE I = 1
.     COMPUTE V = V+1

```



```

.   END WHILE
.   IFTHEN (I NE 1)
.       NEW RECORD IS QUESTTXT (QNLV,ORDERLV)
.       COMPUTE V = (QNLV*100 + ORDERLV)
.       PUT VARS DBNAME FORMNUMB GLOBALNO WORDING
.           VARTEXT1 VARTEXT2 VARTEXT3 VARTEXT4
.           VARTEXT5 VARTEXT6 VARTEXT7 VALTEXT1
.           VALTEXT2
.           = DBNAMELV FORMATLV V WORDGCO
.           VARTXT1 VARTXT2 VARTXT3 VARTXT4
.           VARTXT5 VARTXT6 VARTXT7 VALTXT1
.           VALTXT2
.       END RECORD
.       NEW RECORD IS QUESVAR (SVY,QNLV,ORDERLV)
.       PUT VARS DBNAME FORMNUMB QMNAME WORDING
.           GLOBALNO
.           = DBNAMELV FORMATLV QMNAMELV
.           WORDGCO V
.       END RECORD
.   ENDIF

C ***** ADDING '~~~' FOR VALTEXT' *****
C
.   COMPUTE WRKTXT1 = '~~~'
.   COMPUTE NUMBVT = 2
.   COMPUTE NUMSTR = 1
.   CALL DBASE.MGTDCL

C ***** ADDING '~>' FOR VALTEXT *****
C
.   COMPUTE WRKTXT1 = '~>'
.   COMPUTE NUMBVT = 2
.   COMPUTE NUMSTR = 1
.   CALL DBASE.MGTDCL

C ***** ADDING '~' FOR VALTEXT' *****
C
.   COMPUTE WRKTXT1 = '~'
.   COMPUTE NUMBVT = 2
.   COMPUTE NUMSTR = 1
.   CALL DBASE.MGTDCL

C ***** ADDING '~~~' FOR VARTEXT' *****
C
.   COMPUTE WRKTXT1 = '~~~'
.   COMPUTE NUMBVT = 7
.   COMPUTE NUMSTR = 1
.   CALL DBASE.MGTDCLV

C ***** ADDING '~>' FOR VARTEXT *****

```



```

C
.  COMPUTE WRKTXT1 = '~>'
.  COMPUTE NUMBVT = 7
.  COMPUTE NUMSTR = 1
.  CALL DBASE.MGTDOCV

```

```

C  ***** ADDING '~' FOR VARTEXT' *****
C
.  COMPUTE WRKTXT1 = '~~'
.  COMPUTE NUMBVT = 7
.  COMPUTE NUMSTR = 1
.  CALL DBASE.MGTDOCV

```

RETRIEVAL UPDATE

```

C  ***** LOAD THE INDEX *****
C
.  STRING  *100 RES1,RES2,SCHAIN,STRIND
.  STRING  *8   NAMEIND
.  STRING  *1   CHARVAR
.  INTEGER *4   ORDERIND,V,QNIND
.  INTEGER *2   FINI,RES1B,RES2B
C  -----
.  CIF EQ <CHECKOK>,1
.  READ (<FINDEX>,ERROR= EOF1) STRIND (A80)
.  LOOP
.      STRIND = PACK(TRIMLR(STRIND))
.      COMPUTE SCHAIN = STRIND
.      CALL DBASE.FIRSTSTR
.      COMPUTE NAMEIND = RES1
.      COMPUTE SCHAIN = RES2
.      CALL DBASE.FIRSTSTR
.      COMPUTE QNIND = NUMBR(RES1)
.      COMPUTE SCHAIN = RES2
.      CALL DBASE.FIRSTSTR
.      COMPUTE ORDERIND = NUMBR(RES1)
.      NEW RECORD IS IND (QNIND,ORDERIND)
.      PUT VARS DBNAME = NAMEIND
.      END RECORD
.      READ (<FINDEX>,ERROR= EOF1) STRIND (A80)
.  END LOOP
.  EOF1 : WRITE '----- INDEX TABLE IS CREATED -----'
.  CIF END
END RETRIEVAL

```

RETRIEVAL UPDATE

```

C  ***** LOAD ALL THE VARIABLES *****
C
.  STRING  *100 RES1,RES2,SCHAIN,STRQVAR
.  STRING  *17  QMNAMELV
.  STRING  *8   DBNAMELV
.  STRING  *1   CHARVAR
.  INTEGER *4   ORDERLV,V,QNLV,FORMATLV
.  INTEGER *2   FINI,RES1B,RES2B
C  -----

```



```

. READ (<FQUESVAR>,ERROR= EOF1) STRQVAR (A80)
. LOOP
.   STRQVAR = PACK(TRIMLR(STRQVAR))
.   COMPUTE SCHAIN = STRQVAR
.   CALL DBASE.FIRSTSTR
.   COMPUTE QNLV = NUMBR(RES1)
.   COMPUTE SCHAIN = RES2
.   CALL DBASE.FIRSTSTR
.   COMPUTE ORDERLV = NUMBR(RES1)
.   COMPUTE SCHAIN = RES2
.   CALL DBASE.FIRSTSTR
.   COMPUTE DBNAMELV = RES1
.   COMPUTE SCHAIN = RES2
.   CALL DBASE.FIRSTSTR
.   COMPUTE FORMATLV = NUMBR(RES1)
.   COMPUTE SCHAIN = RES2
.   CALL DBASE.FIRSTSTR
.   COMPUTE QMNAMELV = RES1
.   NEW RECORD IS LISTVAR (QNLV,ORDERLV)
.   PUT VARS DBNAME FORMNUMB QMNAME =
.     DBNAMELV FORMATLV QMNAMELV
.   END RECORD
.   READ (<FQUESVAR>,ERROR=EOF1) STRQVAR (A80)
. END LOOP
. EOF1 : WRITE '----- LISTVAR TABLE IS CREATED -----'
END RETRIEVAL

```

```

C
C MAINPROG
C
C Ce module est le programme principal réalisant le projet
C DOCPROJ. La première étape est de saisir le texte de la
C question et de le stocker après transformation ; la
C deuxième est de constituer la documentation pour chaque
C variable des questions et de la sauver dans la base de
C données définitive.
C
C
C

```

RETRIEVAL UPDATE

```

C -----
C ***** MAIN PROGRAM *****
C
. STRING LENGTH 240
. STRING *100
STRLINE,STRWRK,STRQM,STRCLEAN,SCHAIN,RES1,RES2
. STRING *240 VARTXT1 TO VARTXT7, VALTXT1 TO VALTXT2,
.   WRKTXT1 TO WRKTXT7
. STRING *80 LABEL1 TO LABEL30
. STRING *17 QMNAMELV
. STRING *8 FIRSTVAR, LASTVAR, NAMEIND, DBNAMELV,
.   NPROC1 TO NPROC20, SVY,
.   WRKNAME
. STRING *3 SPECHAR1 TO SPECHAR9
. STRING *1 CHARVAR, POSITION, FIRSTPOS
. INTEGER *4 LENGTH, I, J, K, V, V1, J1, ACOUNT, QCOUNT, HCOUNT,
.   QUESTNO, FORMNO, QNIND, ORDERIND, QWVAR, QNLV,
.   ORDERLV, FORMATLV, NUMVARLV, NUMFORM

```



```

.   INTEGER *2   FINI,RES1B,RES2B,MAXORDER,
                      ERR1 TO ERR130,WORDGCO,GOODPL,NUMSTR,
                      TYPEA1 TO TYPEA43,TYPEB1 TO TYPEB19,
                      TYPEC1 TO TYPEC7

C
C   ----- INITIALISATION OF THE ERRORS -----
C
.   COMPUTE SVY = '<SURVEY>'
.   SET TYPEA1 TO TYPEA43 (1,2,4,6,12,19,22,23,28,29,33,34,35,
                          50,61,62,70,72,75,78,81,85,90,92,
                          93,97,98,101,104,105,106,107,109,
                          110,114,116,121,122,123,124,126,
                          127,128)
.   SET TYPEB1 TO TYPEB19 (9,36,37,43,46,49,60,63,66,69,71,76,
                          77,79,80,108,112,113,115)
.   SET TYPEC1 TO TYPEC7 (57,65,96,100,118,119,120)
.   SET LABE1 TO LABE130 (130 * '.....')

C
C
C
.   COMPUTE LABE1 TO LABE130 (1)   =
.   '   **** WARNING **** : Question without variables.'
.   COMPUTE LABE1 TO LABE130 (4)   =
.   '   **** WARNING **** : Length of A-line too long.'
.   COMPUTE LABE1 TO LABE130 (5)   =
.   '   **** WARNING **** : Length of Q-line too long.'
.   COMPUTE LABE1 TO LABE130 (6)   =
.   '   **** WARNING **** : Length of H-line too long.'
.   COMPUTE LABE1 TO LABE130 (7)   =
.   '   **** WARNING **** : Length of VARTEXT too long.'
.   COMPUTE LABE1 TO LABE130 (8)   =
.   '   **** WARNING **** : Length of VALTEXT too long.'
.   COMPUTE LABE1 TO LABE130 (10)  =
.   '   **** WARNING **** : Strange sequence order in the'
.   '   + 'INDEX.'
.   COMPUTE LABE1 TO LABE130 (11)  =
.   '   **** WARNING **** : Number of variables in QUESVAR'
.   '   + '< in INDEX.'
.   COMPUTE LABE1 TO LABE130 (12)  =
.   '   **** WARNING **** : Number of variables in QUESVAR'
.   '   + '= in INDEX.'
.   COMPUTE LABE1 TO LABE130 (13)  =
.   '   **** WARNING **** : Number of variables in QUESVAR'
.   '   + '> in INDEX.'
.   COMPUTE LABE1 TO LABE130 (14)  =
.   '   **** WARNING **** : Number of variables seems not'
.   '   + 'matching with the format.'
.   COMPUTE LABE1 TO LABE130 (15)  =
.   '   **** WARNING **** : No variable in INDEX.'
.   COMPUTE LABE1 TO LABE130 (16)  =
.   '   **** WARNING **** : No variable in QUESVAR.'
.   COMPUTE LABE1 TO LABE130 (17)  =
.   '   **** WARNING **** : No mean to determine the number'
.   '   + 'of var from format.'
.   COMPUTE LABE1 TO LABE130 (18)  =
.   '   **** WARNING **** : Unknown format.'
.   COMPUTE LABE1 TO LABE130 (101) =
.   '   **** ERROR **** : Not matching - first variable in'
.   '   + 'MASTER and in QUESVAR.'
.   COMPUTE LABE1 TO LABE130 (102) =

```



```

      '      **** ERROR **** : Not matching - last variable in'
      '      + 'MASTER and in QUESVAR.'
.  COMPUTE LABE1 TO LABE130 (103) =
      '      **** ERROR **** : Not matching - format in MASTER'
      '      + 'and in QUESVAR.'
.  COMPUTE LABE1 TO LABE130 (104) =
      '      **** ERROR **** : Not matching - first variable in'
      '      + 'QUESVAR and in INDEX.'
.  COMPUTE LABE1 TO LABE130 (105) =
      '      **** ERROR **** : Not matching - question number in'
      '      + 'QUESVAR and in INDEX.'
.  COMPUTE LABE1 TO LABE130 (106) =
      '      **** ERROR **** : Not matching - last variable in'
      '      + 'QUESVAR and in INDEX.'
.  COMPUTE LABE1 TO LABE130 (108) =
      '      **** ERROR **** : Not matching - first variable in'
      '      + 'MASTER and in INDEX.'
.  COMPUTE LABE1 TO LABE130 (109) =
      '      **** ERROR **** : Not matching - question number in'
      '      + 'MASTER and in INDEX.'
.  COMPUTE LABE1 TO LABE130 (110) =
      '      **** ERROR **** : Not matching - last variable in'
      '      + 'MASTER and in INDEX.'
.  COMPUTE LABE1 TO LABE130 (112) =
      '      **** ERROR **** : Different number of variable in'
      '      + 'MASTER and in INDEX.'
.  COMPUTE LABE1 TO LABE130 (113) =
      '      **** ERROR **** : Question without variable in'
      '      + 'QUESVAR and in INDEX.'

```

```

C  -----
.  COMPUTE ACOUNT = 1
.  COMPUTE HCOUNT = 1
.  COMPUTE QCOUNT = 1
.  SET SPECHAR1 TO SPECHAR9
.  ('~5', '~', '[', ']', '<', '>', '_ $', '$ _', '$')
.  COMPUTE QWVAR = 0
.  CALL DBASE.CLEANTAB
.  COMPUTE POSITION = ' '
.  READ (<FMASTER>, ERROR=EOF1) STRLINE (A80)
.  LOOP
.      COMPUTE FIRSTPOS = SBST(STRLINE, 1, 1)
.      IF ((POSITION NE 'D') AND (FIRSTPOS = 'Q'))
.          COMPUTE WORDGCO = WORDGCO + 1
.      V = LEN(TRIMLR(PACK(STRLINE)))
.      IF ((FIRSTPOS NE ' ') OR (V = 0))
.          COMPUTE POSITION = FIRSTPOS
.      IFTHEN (POSITION = 'D')
.          CALL DBASE.DCATCH
.      ELSEIF (POSITION = 'Q')
.          CALL DBASE.QCATCH
.      ELSEIF (POSITION = 'A')
.          CALL DBASE.ACATCH
.      ELSEIF (POSITION = 'H')
.          CALL DBASE.HCATCH
.      ELSEIF (POSITION = 'C')
.          CALL DBASE.CCATCH
.      ELSEIF (POSITION = 'V')
.          CALL DBASE.VCATCH
.      ENDIF
.      IFTHEN (POSITION = 'V')

```



```

.      PROCESS RECORD LISTVAR WITH (QUESTNO)
.      COMPUTE NUMVARLV = NUMVARLV + 1
.      END RECORD
.      PROCESS RECORD LISTVAR WITH (QUESTNO)
.      GET VARS ORDERLV,DBNAMELV,FORMATLV,
.      QMNAMELV = QORDER,DBNAME,
.      FORMNUMB,QMNAME
.      COMPUTE QNLV = QUESTNO
.      V = SRST(QMNAMELV,'-')
.      IF (V = 0) V = SRST(QMNAMELV,',')
.      IF (V = 0) WRKNAME = SBST(QMNAMELV,1,8)
.      IF (V NE 0) WRKNAME =
.      SBST(QMNAMELV,1,-V-1)
.      IF (FIRSTVAR NE TRIMLR(WRKNAME))
.      COMPUTE ERR1 TO ERR130 (101) = 1
.      IF (V = 0) WRKNAME = ''
.      IF (V NE 0) WRKNAME =
.      SBST(QMNAMELV,-V+1,8)
.      IF (LASTVAR NE TRIMLR(WRKNAME))
.      COMPUTE ERR1 TO ERR130 (102) = 1
.      IF (FORMNO NE FORMATLV)
.      COMPUTE ERR1 TO ERR130 (103) = 1
.      CIF EQ <CHECKOK>,1
.      CALL DBASE.CHECKIND
.      CIF END
.      CALL DBASE.CHFORM
.      CALL DBASE.OUTPUTVA
.      CALL DBASE.INSERTBD
.      END RECORD
.      IFTHEN (NUMVARLV LT 1)
.      COMPUTE ERR1 TO ERR130 (1) = 1
.      COMPUTE QWVAR = QWVAR +1
.      ENDIF
.      CIF EQ <CHECKOK>,1
.      CALL DBASE.MOREIND
.      CIF END
.      CALL DBASE.MGTERR
.      CALL DBASE.CLEANTAB
.      ENDIF
.      READ (<FMASTER>,ERROR=EOF1) STRLINE (A80)
.      END LOOP
.      EOF1 : WRITE '----- MASTER FILE FINISHED -----'
.      CALL DBASE.GLOBANAL
END RETRIEVAL

```

```

C      ***** MANAGE THE TABLE *****
C
C      MANAGEVT
C
C      Ce module a pour objectif d'assembler les différentes
C      parties des lignes nettoyées (Q-lignes, A-lignes ou H-
C      lignes) des questions dans les fichiers de stockage
C      provisoire.
C
C      _____
C
.      COMPUTE GOODPL = 1
.      V = LEN(VARTXT1 TO VARTXT7 (GOODPL))

```



```

. WHILE ((V = 240) AND ((GOODPL+1) LE NUMBVT))
.   COMPUTE GOODPL = GOODPL+1
.   V = LEN(VARTXT1 TO VARTXT7 (GOODPL))
. END WHILE
. V = LEN(VARTXT1 TO VARTXT7 (GOODPL))
. V1 = LEN(STRCLEAN)
. IFTHEN (V LT 240)
.   COMPUTE VARTXT1 TO VARTXT7 (GOODPL) =
.     VARTXT1 TO VARTXT7 (GOODPL) + '~'
. ELSE
.   IF ((GOODPL+1) LE NUMBVT)
.     COMPUTE VARTXT1 TO VARTXT7 (GOODPL+1) = '~'
.   ENDIF
. IFTHEN ((V+1) LT 240)
.   IFTHEN ((V+1+V1) LE 240)
.     COMPUTE VARTXT1 TO VARTXT7 (GOODPL) =
.       VARTXT1 TO VARTXT7 (GOODPL) + STRCLEAN
.   ELSE
.     COMPUTE VARTXT1 TO VARTXT7 (GOODPL) =
.       VARTXT1 TO VARTXT7 (GOODPL) +
.       SBST (STRCLEAN,1,240-V-1)
.     IF ((GOODPL+1) LE NUMBVT)
.       COMPUTE VARTXT1 TO VARTXT7 (GOODPL+1) =
.         SBST(STRCLEAN,240-V,V1-(240-V)+1)
.     ENDIF
.   ELSE
.     IFTHEN ((GOODPL+1) LE NUMBVT)
.       IFTHEN ((V+1) EQ 240)
.         COMPUTE VARTXT1 TO VARTXT7 (GOODPL+1) =
.           STRCLEAN
.       ELSE
.         COMPUTE VARTXT1 TO VARTXT7 (GOODPL+1) =
.           VARTXT1 TO VARTXT7 (GOODPL+1) +
.           STRCLEAN
.       ENDIF
.     ENDIF
.   ENDIF
. ENDIF
. IFTHEN (((GOODPL+1) GT NUMBVT)
.   AND (LEN(VARTXT1 TO VARTXT7(NUMBVT)) = 240))
.   IFTHEN (POSITION = 'A')
.     COMPUTE ERR1 TO ERR130 (4) = 1
.   ELSEIF (POSITION = 'Q')
.     COMPUTE ERR1 TO ERR130 (5) = 1
.   ELSEIF (POSITION = 'H')
.     COMPUTE ERR1 TO ERR130 (6) = 1
.   ENDIF
. ENDIF

```

```

C ***** MANAGE THE VALTEXT *****
C
C

```

```

C MGTDOCL
C

```

```

C Ce module a pour objectif d'ajouter un élément supplé-
C mentaire de documentation à l'attribut 'VALTEXT' d'une
C variable.
C
C
C

```

```

. COMPUTE I=1
. COMPUTE GOODPL = 1
. WHILE ((I LE NUMSTR) AND ((GOODPL+1) LE NUMBVT)
.   AND (LEN(VALTXT1 TO VALTXT2(NUMBVT)) LT 240))
.   COMPUTE GOODPL = 1
.   COMPUTE V=LEN(VALTXT1 TO VALTXT2 (GOODPL))
.   WHILE ((V = 240) AND (GOODPL LE NUMBVT))
.     COMPUTE GOODPL = GOODPL+1
.     COMPUTE V=LEN(VALTXT1 TO VALTXT2 (GOODPL))
.   END WHILE
.   COMPUTE V = LEN(VALTXT1 TO VALTXT2 (GOODPL))
.   COMPUTE V1 = LEN(WRKTXT1 TO WRKTXT7 (I))
.   IFTHEN (V+V1 LE 240)
.     COMPUTE VALTXT1 TO VALTXT2 (GOODPL) =
.       VALTXT1 TO VALTXT2 (GOODPL) +
.       WRKTXT1 TO WRKTXT7 (I)
.   ELSE
.     COMPUTE VALTXT1 TO VALTXT2 (GOODPL) =
.       VALTXT1 TO VALTXT2 (GOODPL) +
.       SBST(WRKTXT1 TO WRKTXT7 (I),1,240-V)
.     IF ((GOODPL+1) LE NUMBVT)
.       COMPUTE VALTXT1 TO VALTXT2 (GOODPL+1) =
.         SBST(WRKTXT1 TO WRKTXT7 (I),
.           240-V+1,V1-(240-V))
.     ENDIF
.     COMPUTE I = I+1
.   END WHILE
. IFTHEN (((GOODPL+1) GT NUMBVT)
.   AND (LEN(VALTXT1 TO VALTXT2 (NUMBVT)) = 240))
.   COMPUTE ERR1 TO ERR130 (8) = 1
. ENDIF

```

C ***** MANAGE THE VARTEXT *****

C
C
C
C
C
C
C
C
C
C
C

MGTDOCV

C Ce module a pour objectif d'ajouter un élément supplé-
C mentaire de documentation à l'attribut 'VARTEXT' d'une
C variable.

C
C
C
C
C
C
C
C
C
C
C

```

. COMPUTE I=1
. COMPUTE GOODPL = 1
. WHILE ((I LE NUMSTR) AND ((GOODPL + 1) LE NUMBVT)
.   AND (LEN(VARTXT1 TO VARTXT7(NUMBVT)) LT 240))
.   COMPUTE GOODPL=1
.   V = LEN(VARTXT1 TO VARTXT7 (GOODPL))
.   WHILE ((V = 240) AND (GOODPL LE NUMBVT))
.     COMPUTE GOODPL = GOODPL+1
.     V = LEN(VARTXT1 TO VARTXT7 (GOODPL))
.   END WHILE
.   V = LEN(VARTXT1 TO VARTXT7 (GOODPL))
.   V1 = LEN(WRKTXT1 TO WRKTXT7 (I))
.   IFTHEN (V+V1 LE 240)
.     COMPUTE VARTXT1 TO VARTXT7 (GOODPL) =
.       VARTXT1 TO VARTXT7 (GOODPL) +
.       WRKTXT1 TO WRKTXT7 (I)

```


C besoin. L'exécution de ce module se déroule après chaque
C question.
C
C
C

```

. COMPUTE NUMIND = 0
. PROCESS RECORD IND WITH (QUESTNO)
.   GET VARS ORDERIND NAMEIND = QORDER DBNAME
.   COMPUTE NUMIND = NUMIND + 1
. END RECORD
. IF ((MAXORDER NE NUMIND) AND (NUMIND NE 0)
.   AND (NUMVARLV NE 0))
.   COMPUTE ERR1 TO ERR130 (10) = 1
.   IFTHEN (NUMBVAR = 1)
.     IF (NUMIND NE 1)
.       COMPUTE ERR1 TO ERR130 (112) = 1
. ELSE
.   IFTHEN (NUMBVAR = 2)
.     IF (NUMIND NE 2)
.       COMPUTE ERR1 TO ERR130 (112) = 1
.   ELSE
.     IFTHEN (NUMBVAR = 3)
.       IF (NUMIND LT 3)
.         COMPUTE ERR1 TO ERR130 (112) = 1
.     ENDIF
.   ENDIF
. ENDIF
. IFTHEN (LEN(FIRSTVAR) NE 0)
.   COMPUTE I = 0
.   PROCESS RECORD IND WITH (QUESTNO)
.     GET VARS ORDERIND NAMEIND = QORDER DBNAME
.     IFTHEN (FIRSTVAR = NAMEIND)
.       COMPUTE I = 1
.     ENDIF
.   END RECORD
.   IF (I = 0)
.     COMPUTE ERR1 TO ERR130 (108) = 1
.   ENDIF
. IF (QUESTNO NE QNIND)
.   COMPUTE ERR1 TO ERR130 (109) = 1
.   IFTHEN (LASTVAR NE '')
.     COMPUTE I = 0
.     PROCESS RECORD IND WITH (QUESTNO)
.       GET VARS ORDERIND NAMEIND = QORDER DBNAME
.       IFTHEN (LASTVAR = NAMEIND)
.         COMPUTE I = 1
.       ENDIF
.     END RECORD
.     IF (I = 0)
.       COMPUTE ERR1 TO ERR130 (110) = 1
.     IF (QUESTNO NE QNIND)
.       COMPUTE ERR1 TO ERR130 (109) = 1
.     ENDIF
.   IF (NUMIND = 0)
.     COMPUTE ERR1 TO ERR130 (15) = 1

```



```
C ***** OUTPUT FILE DOC_LIST *****
```

```
C  
C OUTPUTVA
```

```
C Ce module a pour objectif d'écrire dans le fichier 'Doc_  
C list' la documentation 'VARTEXT' et 'VALTEXT' de chacune  
C des variables même si une erreur s'est produite dans le  
C traitement.
```

```
C  
  
C WRITE ('DOC_LIST') ' '  
C COMPUTE V = -(QNLV*100 + ORDERLV)  
C WRITE ('DOC_LIST') DBNAME LV ' QNLV ' FORMAT LV ' '  
C ORDERLV ' V  
C WRITE ('DOC_LIST') ' QUESTNO ' FORMNO  
C WRITE ('DOC_LIST') '  
C COMPUTE V = 1  
C V1 = LEN(VARTXT1 TO VARTXT7 (V))  
C WHILE ((V LE 7) AND (V1 NE 0))  
C     COMPUTE STRWRK = SBST(VARTXT1 TO VARTXT7 (V),1,80)  
C     WRITE ('DOC_LIST') STRWRK  
C     COMPUTE STRWRK = SBST(VARTXT1 TO VARTXT7 (V),81,80)  
C     WRITE ('DOC_LIST') STRWRK  
C     COMPUTE STRWRK = SBST(VARTXT1 TO VARTXT7 (V),161,80)  
C     WRITE ('DOC_LIST') STRWRK  
C     COMPUTE V = V+1  
C     IF (V LE 7) V1 = LEN(VARTXT1 TO VARTXT7 (V))  
C END WHILE  
C WRITE ('DOC_LIST') ' -----'  
C COMPUTE V = 1  
C V1 = LEN(VALTXT1 TO VALTXT2 (V))  
C WHILE ((V LE 2) AND (V1 NE 0))  
C     COMPUTE STRWRK = SBST(VALTXT1 TO VALTXT2 (V),1,80)  
C     WRITE ('DOC_LIST') STRWRK  
C     COMPUTE STRWRK = SBST(VALTXT1 TO VALTXT2 (V),81,80)  
C     WRITE ('DOC_LIST') STRWRK  
C     COMPUTE STRWRK = SBST(VALTXT1 TO VALTXT2 (V),161,80)  
C     WRITE ('DOC_LIST') STRWRK  
C     COMPUTE V = V+1  
C     IF (V LE 2) V1 = LEN(VALTXT1 TO VALTXT2 (V))  
C END WHILE  
C WRITE ('DOC_LIST') 20T '*****'
```

```
C  
  
C ***** CASE OF Q-LINE *****
```

```
C QCATCH
```

```
C Ce module a pour objectif de saisir une Q-ligne provenant  
C du Master File, de la nettoyer des caractères de contrôle  
C de QUESTMAST et de la stocker provisoirement dans le  
C fichier 'qtable'.  
  
C  
C CALL DBASE.CLEANER  
C COMPUTE STRCLEAN = STROM
```



```

.   SET VARTXT1 TO VARTXT7 (7 * '')
.   IFTHEN (FIRSTPOS = 'Q')
.       RECORD IS QTABLE (QCOUNT)
.           IF (LEN(STRCLEAN) GE 2)
.               COMPUTE VARTXT1 =
.                   SBST(STRCLEAN,2,LEN(STRCLEAN)-1)
.               PUT VARS QLINE1 QLINE2 QLINE3 QLINE4 QLINE5
.                   QLINE6 QLINE7 = VARTXT1 VARTXT2 VARTXT3
.                   VARTXT4 VARTXT5 VARTXT6 VARTXT7
.           END RECORD
.       COMPUTE QCOUNT = QCOUNT+1
.   ELSE
.       COMPUTE I = QCOUNT - 1
.       RECORD IS QTABLE (I)
.           GET VARS VARTXT1 TO VARTXT7 = QLINE1 TO QLINE7
.           COMPUTE NUMBVT = 7
.           CALL DBASE.MANAGEVT
.           PUT VARS QLINE1 TO QLINE7 = VARTXT1 VARTXT2
.                                       VARTXT3 VARTXT4
.                                       VARTXT5 VARTXT6
.                                       VARTXT7
.       END RECORD
.   ENDIF

```

```

C   ***** Q-LINE FOR VARTEXT *****
C
.   COMPUTE J = ORDERLV
.   WHILE (J LE (QCOUNT - 1))
.       RECORD IS QTABLE (J)
.           IFTHEN (J NE ORDERLV)
.               CALL DBASE.JOINVW
.           ENDIF
.       GET VARS WRKTXT1 TO WRKTXT7 = QLINE1 TO QLINE7
.       COMPUTE NUMBVT = 7
.       COMPUTE NUMSTR = 7
.       CALL DBASE.MGTDOCV
.   END RECORD
.   COMPUTE J = J + NUMVARLV
.   END WHILE

```

```

C   ***** SUITE OF A-LINE FOR VALTEXT *****
C
.   COMPUTE J1 = (ACOUNT-1)/WORDGCO
.   COMPUTE J = 1
.   RECORD IS ATABLE (J)
.       GET VARS WRKTXT1 TO WRKTXT2 = ALINE1 TO ALINE2
.       COMPUTE NUMBVT = 2
.       COMPUTE NUMSTR = 2
.       CALL DBASE.MGTDOCL
.   END RECORD
.   COMPUTE J = J+1
.   WHILE (J LE (ACOUNT-1))
.       IFTHEN ((MOD (J,J1) = 1) OR (J1 = 1 ))
.           CALL DBASE.JOINLW
.       ELSE
.           CALL DBASE.JOINLX

```



```

.      ENDIF
.      RECORD IS ATABLE (J)
.          GET VARS WRKTXT1 TO WRKTXT2 = ALINE1 TO ALINE2
.          COMPUTE NUMSTR = 2
.          CALL DBASE.MGTDACL
.      END RECORD
.      COMPUTE J = J+1
.  END WHILE

```

```

C ***** SUITE OF A-LINE FOR VARTEXT *****
C
.  COMPUTE J1 = (ACOUNT-1)/WORDGCO
.  COMPUTE J = 1
.  RECORD IS ATABLE (J)
.      GET VARS WRKTXT1 TO WRKTXT2 = ALINE1 TO ALINE2
.      COMPUTE NUMBVT = 7
.      COMPUTE NUMSTR = 2
.      CALL DBASE.MGTDACV
.  END RECORD
.  COMPUTE J = J+1
.  WHILE (J LE (ACOUNT-1))
.      IFTHEN ((MOD (J,J1) = 1) OR (J1 = 1))
.          CALL DBASE.JOINVW
.      ELSE
.          CALL DBASE.JOINVX
.      ENDIF
.      RECORD IS ATABLE (J)
.          GET VARS WRKTXT1 TO WRKTXT2 = ALINE1 TO ALINE2
.          COMPUTE NUMSTR = 2
.          CALL DBASE.MGTDACV
.      END RECORD
.      COMPUTE J = J+1
.  END WHILE

```

```

C ***** SUITE OF H-LINE FOR VALTEXT *****
C
.  COMPUTE J1 = (HCOUNT-1)/WORDGCO
.  COMPUTE J = 1
.  RECORD IS HTABLE (J)
.      GET VARS WRKTXT1 TO WRKTXT2 = HLINE1 TO HLINE2
.      COMPUTE NUMBVT = 2
.      COMPUTE NUMSTR = 2
.      CALL DBASE.MGTDACL
.  END RECORD
.  COMPUTE J = J+1
.  WHILE (J LE (HCOUNT-1))
.      IFTHEN ((MOD (J,J1) = 1) OR (J1 = 1))
.          CALL DBASE.JOINLW
.      ELSE
.          CALL DBASE.JOINLX
.      ENDIF
.      RECORD IS HTABLE (J)
.          GET VARS WRKTXT1 TO WRKTXT2 = HLINE1 TO HLINE2
.          COMPUTE NUMSTR = 2
.          CALL DBASE.MGTDACL
.      END RECORD

```



```

.      COMPUTE J = J+1
.      END WHILE

C      ***** SUITE OF H-LINE FOR VARTEXT *****
C
.      COMPUTE J1 = (HCOUNT-1)/WORDGCO
.      COMPUTE J = 1
.      RECORD IS HTABLE (J)
.          GET VARS WRKTXT1 TO WRKTXT2 = HLINE1 TO HLINE2
.          COMPUTE NUMBVT = 7
.          COMPUTE NUMSTR = 2
.          CALL DBASE.MGTDOCV
.      END RECORD
.      COMPUTE J = J+1
.      WHILE (J LE (HCOUNT-1))
.          IFTHEN ((MOD (J,J1) = 1) OR (J1 = 1))
.              CALL DBASE.JOINWV
.          ELSE
.              CALL DBASE.JOINWV
.          ENDIF
.          RECORD IS HTABLE (J)
.              GET VARS WRKTXT1 TO WRKTXT2 = HLINE1 TO HLINE2
.              COMPUTE NUMSTR = 2
.              CALL DBASE.MGTDOCV
.          END RECORD
.          COMPUTE J = J+1
.      END WHILE

C      ***** SUITE OF Q-LINE FOR VARTEXT *****
C
.      COMPUTE J1 = (QCOUNT-1)/WORDGCO
.      COMPUTE J = 1
.      RECORD IS QTABLE (J)
.          GET VARS WRKTXT1 WRKTXT2 WRKTXT3 WRKTXT4 WRKTXT5
.              WRKTXT6 WRKTXT7 = QLINE1 QLINE2 QLINE3
.              QLINE4 QLINE5 QLINE6 QLINE7
.          COMPUTE NUMBVT = 7
.          COMPUTE NUMSTR = 7
.          CALL DBASE.MGTDOCV
.      END RECORD
.      COMPUTE J = J+1
.      WHILE (J LE (QCOUNT-1))
.          IFTHEN ((MOD (J,J1) = 1) OR (J1 = 1))
.              CALL DBASE.JOINWV
.          ELSE
.              CALL DBASE.JOINWV
.          ENDIF
.          RECORD IS QTABLE (J)
.              GET VARS WRKTXT1 WRKTXT2 WRKTXT3 WRKTXT4
.                  WRKTXT5 WRKTXT6 WRKTXT7 = QLINE1
.                  QLINE2 QLINE3 QLINE4 QLINE5 QLINE6
.                  QLINE7
.              COMPUTE NUMSTR = 7
.              CALL DBASE.MGTDOCV
.          END RECORD
.          COMPUTE J = J+1

```


. END WHILE

C ***** ADDING 'TICKED-NOT TICKED' *****

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

TICKED

Ce module a pour objectif d'ajouter dans l'attribut VALTEXT de la variable traitée le texte suivant 'Ticked
~~Not ticked'.

```
. COMPUTE WRKTXT1 = 'TICKED~~NOT TICKED'
. COMPUTE NUMBVT = 2
. COMPUTE NUMSTR = 1
. CALL DBASE.MGTDCL
```

C ***** CASE IF V-LINE *****

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

VCATCH

Ce module a pour objectif de saisir le groupe de variables associé à la question. La source en est le fichier Master File.

```
. COMPUTE STRCLEAN = STRLINE
. FOR V1 = 1,2,1
.   V = SRST(STRCLEAN,'=')
.   COMPUTE V = (-V)
.   STRCLEAN = SBST(STRCLEAN,V+1,LEN(STRCLEAN)-V)
. END FOR
. V = SRST(STRCLEAN,'RANGE')
. COMPUTE V = (-V)
. STRCLEAN = PACK(TRIMLR(SBST(STRCLEAN,1,V-1)))
. IFTHEN ((PATTERN(STRCLEAN,'@-@') EQ 1) OR
.         (PATTERN(STRCLEAN,'@,@') EQ 1))
.   V = SRST(STRCLEAN,'-')
.   IFTHEN (V NE 0)
.     COMPUTE NUMBVAR = 3
.   ELSE
.     V = SRST(STRCLEAN,',')
.     IF (V NE 0)
.       COMPUTE NUMBVAR = 2
.   ENDIF
.   COMPUTE V = (-V)
.   FIRSTVAR = SBST(STRCLEAN,1,V-1)
.   LASTVAR = SBST(STRCLEAN,V+1,LEN(STRCLEAN)-V)
. ELSE
.   IFTHEN (LEN(TRIMLR(STRCLEAN)) NE 0)
.     COMPUTE FIRSTVAR = TRIMLR(STRCLEAN)
.     COMPUTE LASTVAR = ''
.     COMPUTE NUMBVAR = 1
.   ELSE
.     COMPUTE FIRSTVAR = ''
```



```

.      COMPUTE LASTVAR = ''
.      COMPUTE NUMBVAR = 0
.      ENDIF
.  ENDIF

C ***** WRITE THE ERRORS *****
C
C WRITEERR
C
C Ce module a pour objectif d'écrire dans le fichier 'Doc_
C err' l'ensemble des erreurs et warnings qui se sont
C produits lors de l'exécution du programme.
C
C
C
C
C WRITE ('DOC_ERR') ' '
C WRITE ('DOC_ERR') 'QUESTION NUMBER : 'QUESTNO
C WRITE ('DOC_ERR') 'FORMAT NUMBER : 'FORMNO
C WRITE ('DOC_ERR') ' '
C PROCESS RECORD LISTVAR WITH (QUESTNO)
C     GET VARS ORDERLV DBNAMELV FORMATLV QMNAMELV =
C         QORDER DBNAME FORMNUMB QMNAME
C     WRITE ('DOC_ERR') QUESTNO ' ' ORDERLV ' '
C         DBNAMELV ' ' FORMATLV ' '
C         QMNAMELV
C
C END RECORD
C WRITE ('DOC_ERR') ' '
C CIF EQ <CHECKOK>,1
C PROCESS RECORD IND WITH (QUESTNO)
C     GET VARS ORDERIND NAMEIND = QORDER DBNAME
C     WRITE ('DOC_ERR') QUESTNO ' ' ORDERIND ' '
C         NAMEIND
C
C END RECORD
C CIF END
C WRITE ('DOC_ERR') ' '
C COMPUTE I = 1
C WHILE (I LE 130)
C     COMPUTE V = ERR1 TO ERR130 (I)
C     IFTHEN (V = 1)
C         COMPUTE STRCLEAN = LABE1 TO LABE130 (I)
C         WRITE ('DOC_ERR') STRCLEAN
C         IF (I = 14) WRITE ('DOC_ERR') '
C             'Number of variables expected : '
C             NUMFORM
C     ENDIF
C     COMPUTE I = I + 1
C END WHILE

C ***** ADDING 'YES-NO' *****
C
C YESNO
C
C Ce module a pour objectif d'ajouter dans l'attribut
C VALTEXT de la variable traitée le texte suivant 'YES~~NO'.
C
C
C

```



```
C
.  COMPUTE WRKTXT1 = 'YES~~NO'
.  COMPUTE NUMBVT = 2
.  COMPUTE NUMSTR = 1
.  CALL DBASE.MGTDOCL
```



```

C ***** LIST ALL THE VARIABLES *****
C
C LLISTVAR
C
C Ce programme a pour objectif de lister dans le fichier
C 'DOC LV' le contenu de la table LISTVAR de la base de
C données DOCPROJ.
C
C
C
C RETRIEVAL
C . PROCESS REC LISTVAR
C . GET VARS ALL
C . WRITE (DOC_LV) QMNUMBR QORDER DBNAME FORMNUMB QMNAME
C . END REC
C END RETRIEVAL

```

```

C ***** LIST ALL THE INDEX *****
C
C LIND
C
C Ce programme a pour objectif de lister dans le fichier
C 'DOC_IND' le contenu de table IND de la base de données
C DOCPROJ.
C
C
C
C RETRIEVAL
C . PROCESS REC IND
C . GET VARS ALL
C . WRITE (DOC_IND) QMNUMBR QORDER DBNAME
C . END REC
C END RETRIEVAL

```

```

C ***** LIST ALL THE PROCEDURES *****
C
C LPROC
C
C Ce programme a pour objectif de lister dans le fichier
C 'DOC_PROC' le contenu de la table NAMEPROC de la base de
C données DOCPROJ.
C
C
C
C RETRIEVAL
C . STRING *8 WRKNAME
C . INTEGER *4 V,V1,WRK
C . COMPUTE WRK = 0
C . PROCESS REC NAMEPROC
C . GET VARS ALL
C . IFTHEN (CHFORMAT NE WRK)
C . WRITE ('DOC_PROC') 'FORMAT NUMBER : ' CHFORMAT
C . WRITE ('DOC_PROC') '-----'
C . WRITE ('DOC_PROC') ' '
C . ENDIF

```



```

.      COMPUTE WRK = CHFORMAT
.      IFTHEN (LEN(PROC1) NE 0)
.          IFTHEN (TYPE = 'A')
.              WRITE ('DOC_PROC') 'Operations to execute in any'
.                                      'cases'
.
.          ENDIF
.          IFTHEN ((TYPE = 'C') AND (ORDER = 1))
.              WRITE ('DOC_PROC') 'Operations to execute only'
.                                      'for the FIRST variable of'
.                                      'the variable''s set'
.
.          ENDIF
.          IFTHEN ((TYPE = 'C') AND (ORDER = 3))
.              WRITE ('DOC_PROC') 'Operations to execute only'
.                                      'for the OTHER variables'
.
.          ENDIF
.          IFTHEN ((TYPE = 'B') AND (ORDER = 1))
.              WRITE ('DOC_PROC') 'Operations to execute only'
.                                      'if there is ONE variable'
.
.          ENDIF
.          IFTHEN ((TYPE = 'B') AND (ORDER = 3))
.              WRITE ('DOC_PROC') 'Operations to execute only'
.                                      'if there are MORE than one'
.                                      'variable'
.
.          ENDIF
.          COMPUTE V = 1
.          V1 = LEN(PROC1 TO PROC20(V))
.          WHILE (V1 NE 0)
.              COMPUTE WRKNAME = PROC1 TO PROC20(V)
.              WRITE ('DOC_PROC') 20T WRKNAME
.              COMPUTE V = V + 1
.              V1 = LEN(PROC1 TO PROC20(V))
.          END WHILE
.      ENDIF
.      WRITE ('DOC_PROC') ' '
.  END REC
END RETRIEVAL

```

```

C      ***** LIST ALL THE INDEX QUESVAR *****
C
C      LQUESVAR
C
C      Ce programme a pour objectif de lister dans le fichier
C      'DOC_VAR' le contenu de la table QUESVAR de la base de
C      données DOCPROJ.
C
C      _____
C
RETRIEVAL
.      PROCESS REC QUESVAR
.      GET VARS ALL
.      WRITE (DOC_VAR) IDSUR QMNUMBR QORDER DBNAME FORMNUMB
.                               QMNAME WORDING GLOBALNO
.
.      END REC
END RETRIEVAL

```



```

C ***** LIST ALL THE VARTEXT AND VALTEXT *****
C
C LQUESTXT
C
C Ce programme a pour objectif de lister dans le fichier
C 'DOC_QUESTION' le contenu de la table QUESTTXT de la base
C de données DOCPROJ.
C
C
C
C
C RETRIEVAL
C . STRING LENGTH 240
C . STRING *80 STRWRK
C . INTEGER *4 V
C . PROCESS REC QUESTTXT
C . GET VARS ALL
C . WRITE (DOC_QUESTION) QMNUMBR QORDER DBNAME FORMNUMB GLOBALNO
C   WORDING
C . WRITE (DOC_QUESTION)
C . COMPUTE V = 1
C . V1 = LEN(VARTEXT1 TO VARTEXT7 (V))
C . WHILE ((V LE 7) AND (V1 NE 0))
C .   COMPUTE STRWRK = SBST(VARTEXT1 TO VARTEXT7 (V),1,80)
C .   WRITE ('DOC_QUESTION') STRWRK
C .   COMPUTE STRWRK = SBST(VARTEXT1 TO VARTEXT7 (V),81,80)
C .   WRITE ('DOC_QUESTION') STRWRK
C .   COMPUTE STRWRK
C .     = SBST(VARTEXT1 TO VARTEXT7 (V),161,80)
C .   WRITE ('DOC_QUESTION') STRWRK
C .   COMPUTE V = V+1
C .   IF (V LE 7) V1 = LEN(VARTEXT1 TO VARTEXT7 (V))
C . END WHILE
C . WRITE ('DOC_QUESTION') ' -----
C
C . COMPUTE V =1
C . V1 = LEN(VALTEXT1 TO VALTEXT2 (V))
C . WHILE ((V LE 2) AND (V1 NE 0))
C .   COMPUTE STRWRK = SBST(VALTEXT1 TO VALTEXT2 (V),1,80)
C .   WRITE ('DOC_QUESTION') STRWRK
C .   COMPUTE STRWRK = SBST(VALTEXT1 TO VALTEXT2 (V),81,80)
C .   WRITE ('DOC_QUESTION') STRWRK
C .   COMPUTE STRWRK
C .     = SBST(VALTEXT1 TO VALTEXT2 (V),161,80)
C .   WRITE ('DOC_QUESTION') STRWRK
C .   COMPUTE V = V+1
C .   IF (V LE 2) V1 = LEN(VALTEXT1 TO VALTEXT2 (V))
C . END WHILE
C . WRITE ('DOC_QUESTION') 20T '*****'
C . END RECORD
C END RETRIEVAL

```


ANNEXE 6 : Application de la notion de raisonnement déductif¹

Nous avons vu au chapitre 4, section 4.2 que QUESTMAST produisait le schéma de base de données suivant :

Record Schema 1

Data List	Fixed (1)		
/1	Caseld	6-11	(1)
/1	Qultecol	12-13	(1)
/1	Pqultte1	14-15	(1)
/1	Pqultte2	16-17	(1)
/1	Pqultte3	18-19	(1)
/1	Regret	20-21	(1)

Var Ranges

Qultecol	(1 4) /
Pqultte1	(1 2) /
Pqultte2	(1 2) /
Pqultte3	(1 2) /
Regret	(1 3) /

End Schema 1

Comment QUESTMAST crée-t'il ce schéma ?

Chaque question a un nom et un format. Le format définit :

1. comment la question sera représenté sur la page,
2. le nombre de variables associées à la question,
3. les bornes de ces variables.

Par exemple, toutes les questions de type "cocher tout ce qui convient" ont le même nombre de variables (en fonction du nombre de lignes de réponse) et ont des bornes de 1-2.

Les noms des variables sont déduites du nom de la question (ex: la question "Pourquoi avez-vous quitté l'école ?" a les

¹ Inspiré de "Deductive Reasoning in Questionnaire Design" de J.M. LAMB, CES, 1990.

variables Pquitte1.. Pquitte3) ou sont définies par l'utilisateur.

La question n°1 de l'exemple précédent, portant le nom de Quittecol, possède le numéro de format 22 (dans la dernière version de QUESTMAST). Les caractéristiques de ce format sont les suivantes :

- * nombre de variable : 1 ;
- * borne : 1-4 ;
- * valeur manquante : 99.

Pour QUESTMAST, ce format 22 n'est pas différent d'un autre format. Mais pour les chercheurs, ce format est essentiel car il traduit la notion de question-filtre. Par souci de complétude, il serait dès lors bon de rajouter aussi cette notion dans les caractéristiques de définition des formats dans QUESTMAST :

- * filtre : oui.

Un des inconvénients du système actuel est que, pour chaque question-filtre, on est obligé d'ajouter du texte explicatif à la variable associée, comme par exemple :

- si non, aller à la page suivante ;
- si non, aller à la question 10 ;
- si non, aller à la page 11.

Bien évidemment, il serait plus audacieux d'effectuer ce saut non pas vers une place physique sur la page, mais vers une question logique. Ceci pourrait se faire en spécifiant un nom de question.

Cependant cette nouvelle caractéristique ajoutée à celles déjà existantes (à savoir la notion de filtre) requiert qu'un nom de "saut".

Depuis que les variables-filtres ont été considérées comme essentielles aux yeux des chercheurs, on est arrivé à essayer de déduire la réponse. Il est à remarquer qu'assez souvent les répondants omettent la question-filtre, mais répondent

tout de même aux questions qui dépendent de ce filtre. D'où la nécessité d'introduire des valeurs manquantes :

- 3 signifie 'Codeur impliquant oui' ;
- 99 signifie 'pas de réponse' ;
- 4 signifie 'a répondu non mais
a impliqué oui ailleurs'.

Voici une méthode qui permettrait de palier à ces problèmes de valeurs manquantes : les questionnaires sont décrits pour QUESTMAST en termes de variables ayant un nom. Donc, ces variables peuvent être décrites par la séquence suivante :

```

Quitecol
|
Pquitte
|
Regret


```

Si on ajoute la notion de question-filtre, on obtient alors la séquence :

```

Quitecol
Pquitte
Regret

```



Et de là, QUESTMAST pourrait déduire que Pquitte est dépendant de Quitecol.

Qu'est-ce que cela nous apporte ? Si les chercheurs du CES adoptent le fait que la valeur manquante 96 est attribuée pour les variables dépendant d'un filtre, on peut établir la vérification suivante :

Si Quitecol = 2
Alors Pquitte1...Pquitte3 = 96 (1)

Il serait intéressant d'introduire un nouveau concept : celui d'"information dépendant du type de la question". Ceci pourrait se combiner comme suit :

Si (la question est un filtre) et
 (la valeur de cette question = 2)
 Alors (la valeur manquante
 des questions dépendantes = 96)

Ce concept est d'un niveau plus élevé que celui défini en (1). On pourrait encore définir une règle plus générale du type :

Si (<filtre> = 2)
 Alors (<quest> = <valeur>)
 où <valeur> serait, par exemple, 96
 <quest> représente les questions dépendantes

Bien sûr, en raisonnant en termes de méta-données, il serait bon d'interpréter la notion de valeur manquante et de générer alors des règles telles que :

Soit f un filtre
 Et {d(f)} l'ensemble des questions (variables)
 dépendantes de f

Si une de ces variables a une valeur manquante, on l'identifiera par la notion de "non-réponse". Si toutes les variables de l'ensemble ont la même valeur manquante, l'ensemble est dit avoir la valeur manquante "non-réponse".

Si l'on définit les valeurs manquantes suivantes :

<VM1> signifiant 'pas répondu'
 <VM2> signifiant 'question non pertinente'
 <IO> signifiant 'implique oui'

On peut alors construire deux autres règles :

- * Si ({d(f)} est l'ensemble des valeurs manquantes <VM1>)
 et (f prend la valeur 2)
 Alors ({d(f)} prend la valeur manquante <VM2>)
- * Si non ({d(f)} est l'ensemble des valeurs manquantes
 <VM1>)
 Alors (filtre = <IO>)

Il est à remarquer que la construction d'une base de données pour chaque étude ne dépend pas uniquement du contenu

du questionnaire. En plus du contenu, on peut relever trois autres facteurs :

- * la pratique d'encodage ;
- * les détails administratifs ;
- * les facteurs du monde réel que l'étude en question tente de cerner.

Ces trois facteurs sont difficiles à saisir pour documenter une étude. Si l'on prend les exemples développés précédemment, on serait tenté d'y voir une approche 'Expert System' qui exigerait alors une définition bien plus rigoureuse de Quitecol et qui, à partir de cette définition, effectuerait les vérifications de compatibilité. En voici quelques exemples :

- * Si Quitecol = 1 ou 3
Alors l'élève a quitté l'école

(renseignement extrait du questionnaire)
- * Si l'élève a quitté l'école
Alors l'âge de l'élève est > 17

(renseignement déduit du monde réel)
- * Age de l'élève = date de retour du
questionnaire - date de naissance

(renseignement obtenu de l'administration)

BIBLIOGRAPHIE

BRYANT I., BURNHILL P.M., LAMB J.M., RAFFE D., "Report on the 1984 pilot of the Scottish Young People's Survey", Centre for Educational Sociology, University of Edinburgh, 1985.

FAIRGRIEVE J., "Processing a Survey using Metadata Techniques", Centre for Educational Sociology, University of Edinburgh, 1991.

FAIRGRIEVE J., LAMB J.M., "Scottish Young People's Survey 1989 : Technical Report", Centre for Educational Sociology, University of Edinburgh, 1990.

LAMB J.M., "Deductive Reasoning in Questionnaire Design", Centre for Educational Sociology, University of Edinburgh, 1990.

LAMB J.M., MIDDLETON L., "Techniques and software for the processing of a large postal survey", The Statistician, 40(2), pp.139-143, 1991.

LAMB J.M., SHAW A., FAIRGRIEVE J., "The Scottish Young People's Survey : Documentation of the 1989 Spring Database", Centre for Educational Sociology, University of Edinburgh, 1991.

LAMB J.M., "QUESTMAST: a package to aid the design and construction of questionnaires", Social Science Research Council Survey Archive Bulletin, S3-S4, 1981.

RITCHIE P., "Design of Conceptual Model for Documentation Database and Prototype in SIR : Final Report" The Economic and Social Research Council on Project No. R000231293, Centre for Educational Sociology, University of Edinburgh, 1991.

RITCHIE P., "ESRC DocDb Project", Centre for Educational Sociology, University of Edinburgh, Oct.1989.

BODART F., PIGNEUR Y., "Conception assistée des systèmes d'information (Méthode, Modèles, Outils)", Ed. Masson, 1989.

HAINAUT J.-L., "Conception assistée des applications informatiques (Conception de la base de données)", Ed. Masson, 1986.

JAVEAU C., "L'enquête par questionnaire : le manuel à l'usage du praticien", Ed. de l'Université de Bruxelles, 1978.

MUCCHELLI R., "Le questionnaire dans l'enquête psycho-sociale", Entreprise Moderne d'Édition ESF, 1978.

D'HAINAUT L., "Concepts et Méthodes de la Statistique (tome 1)", Ed. Labor, 1975.

ZIKMUND W.G., "Business Research Methods", Library of Congress (Third Edition), 1991.

COCHRAN W.G., "Sampling Techniques", A Wiley Publication in Applied Statistics (third Edition), 1977.